Calibrating Deep Convolutional Gaussian Processes

G.-L. Tran[1] **E. V. Bonilla**[2] **J. P. Cunningham**[3] **P. Michiardi**[1] **M. Filippone**[1]
1 - EURECOM, France 2 - CSIRO's Data61 and UNSW 3 - Columbia University, USA

Abstract

The wide adoption of Convolutional Neural Networks (CNNs) in applications where decision-making under uncertainty is fundamental, has brought a great deal of attention to the ability of these models to accurately quantify the uncertainty in their predictions. Previous work on combining CNNs with Gaussian processes (GPs) has been developed under the assumption that the predictive probabilities of these models are well-calibrated. In this paper we show that, in fact, current combinations of CNNs and GPs are miscalibrated. We propose a novel combination that considerably outperforms previous approaches on this aspect, while achieving state-of-the-art performance on image classification tasks.

1 Introduction

The wide adoption of Convolutional Neural Networks (CNNs) in increasingly popular pieces of technology such as self driving cars and medical imaging, where decision-making under uncertainty is fundamental, has brought attention to the ability of these learning architectures to accurately quantify the uncertainty in their predictions (Kendall and Gal, 2017; Gal and Ghahramani, 2016b). In short, the reliability of predictive probabilities of learning algorithms can be evaluated through the analysis of their calibration (Flach, 2016). In particular, a classifier is well calibrated when its output offers an accurate account of the probability of a given class, i.e. when it predicts a given class label with probability p that matches the true proportion p of test points belonging to that class.

The calibration properties of standard classifiers and neural networks have been studied in the literature (Kull et al., 2017; Niculescu-Mizil and Caruana, 2005), which has shown that classifiers that use the standard cross-entropy loss are generally well calibrated. Perhaps surprisingly, modern CNNs, which are a particular case of deep neural networks (DNNs), have been found to be miscalibrated, and the depth of convolutional filters is the main factor affecting calibration (Guo et al., 2017). The work in Guo et al. (2017) shows that regularization, implemented through weight decay, improves calibration and that, ultimately, simple methods such as post-calibration (Platt, 1999) can be an effective remedy for most calibration issues of CNNs.

Alternatively, Bayesian CNNs (Gal and Ghahramani, 2016b) where convolutional filters are inferred using Bayesian inference techniques, seem like perfect candidates to model uncertainty in these architectures in a principled way. However, while Bayesian CNNs have been shown to be effective in obtaining state-of-the-art performance in image classification tasks, we are not aware of studies that show their calibration properties. Hence, our first contribution is to investigate the calibration properties of Bayesian CNNs.

Along a similar vein, independently of the works on Bayesian CNNs, there have been other attempts to give a probabilistic flavor to CNNs by combining them with Gaussian processes (GPs, (Rasmussen and Williams, 2006)). Most of these approaches can be seen as a way to parameterize a CNN-based covariance for GPs, and the aim is to learn end-to-end both the filters and the GPs (see, e.g., Bradshaw et al. (2017); Wilson et al. (2016)). A crucial aspect that the literature has overlooked, however, is that methods that combine CNNs and GPs suffer from the same issues of miscalibration that characterize modern CNNs. Therefore, the second contribution of this paper is to show that current combinations of CNNs and GPs are miscalibrated.

Consequently, as our third contribution, we propose a novel combination of CNNs and GPs that is indeed well-calibrated, while being simple to implement. In particular, we propose to replace the fully connected layers of CNNs with GPs that we approximate with random features (Cutajar et al., 2017; Lázaro-Gredilla et al., 2010). Due to this approximation, the resulting model

Proceedings of the 22[nd] International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

becomes a Bayesian CNN with a nonlinear transformation applied to the convolutional features. Building on the connection between variational inference and dropout, we apply Monte Carlo dropout (MCD, (Gal and Ghahramani, 2016a)) to carry out joint inference over the filters and the approximate GPs, thus obtaining an end-to-end learning method for the proposed model, which we call CNN+GP(RF). The resulting approach is characterized by a number of attractive features: (i) it is well calibrated, given that it uses the multinomial likelihood and the filters are regularized using Bayesian inference techniques; (ii) it is as scalable as state-of-the-art CNNs, in so much as it can be trained using mini-batch updates and can exploit GPU and distributed computing; (iii) unlike other works that combine CNNs and GPs, it is as easy to implement as standard CNNs, as it leverages the equivalence of GPs approximated with random features and Bayesian DNNs (Cutajar et al., 2017; Gal and Turner, 2015; Neal, 1996), and the connections between dropout and variational inference (Gal and Ghahramani, 2016a). We extensively validate these properties in a variety of image classification tasks.

Our final contribution extends the above framework by replacing the last layer of CNNs with Deep GPs (Cutajar et al., 2017) and by proposing the use of structured random features to obtain faster and more compact GP approximations (Le et al., 2013; Yu et al., 2016). In all, our proposal considerably improves on classification accuracy compared to previous combinations of CNNs and GPs (e.g., ∼88% on CIFAR10 and ∼67% on CIFAR100, all without data augmentation), while being competitive with state-of-the-art CNNs; we are not aware of other GP works that approach these results. Crucially, we achieve these performance without compromising on calibration, again considerably improving on previous approaches that combine CNNs and GPs.

2 Related Work

Calibration of Convolutional Networks: The issue of calibration of classifiers in machine learning was popularized in the 90's with the use of support vector machines for probabilistic classification (Platt, 1999). Calibration techniques aim to learn a transformation of the output using a validation set in order for the transformed output to give a reliable account of the actual probability of class labels (Flach, 2016); interestingly, calibration can be applied regardless of the probabilistic nature of the untransformed output of the classifier. Popular calibration techniques include Platt scaling (Platt, 1999) and isotonic regression (Zadrozny and Elkan, 2002).

Classifiers based on Deep Neural Networks (DNNs) have been shown to be well-calibrated (Niculescu-Mizil and Caruana, 2005). The reason is that the optimization of the cross-entropy loss promotes calibrated output. The same loss is used in Platt scaling and it corresponds to the correct multinomial likelihood for class labels. Recent sudies on the calibration of CNNs, which are a particular case of DNNs, however, show that depth has a negative impact on calibration, despite the use of a cross-entropy loss, and that regularization improves the calibration properties of classifiers (Guo et al., 2017).

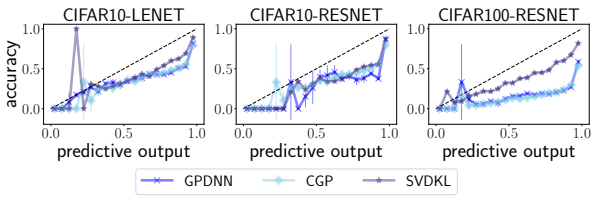

Figure 1: Reliability diagrams for three state-of-the-art combinations of CNNs and GPs, i.e GPDNN (Bradshaw et al., 2017), CGP (van der Wilk et al., 2017), SVDKL (Wilson et al., 2016) applied to CIFAR10 and CIFAR100 data sets with LeNet and RESNET architectures. See table 1 for details on the convolutional architectures that we apply to CIFAR10 and CIFAR100. Because it is not possible to specify the convolutional structure in CGP (van der Wilk et al., 2017), the left and central panels show the same curve for CGP.

Combinations of Conv Nets and Gaussian Processes: Thinking of Bayesian priors as a form of regularization, it is natural to assume that Bayesian CNNs can "cure" the miscalibration of modern CNNs. Despite the abundant literature on Bayesian DNNs (Neal, 1996; Mackay, 1994), far less attention has been devoted to Bayesian CNNs (Gal and Ghahramani, 2016a), and the calibration properties of these approaches have not been investigated.

Several approaches have proposed the combination of CNNs and GPs as a means to give a probabilistic character to CNNs. Most of these works are based on ideas developed in the context of manifold GPs (Calandra et al., 2016), where inputs are transformed using some parametric transformation. In these works, the parametric transformation is based on convolutional layers, and scalability to large data is achieved through the use of ideas drawn from the literature on scalable GPs, for example the Stochastic Variational Deep Kernel Learning (SVDKL) approach in Wilson et al. (2016). In contrast, the work on hybrid GPs and DNNs (GPDNN, (Bradshaw et al., 2017)) combines CNNs and GPs using an inducing point approximation. Other recent approaches that aim to introduce convolutions in the calculation of the covariance between images include

the work in van der Wilk et al. (2017), which proposes a way to construct covariances between domains/patches, mimicking the computations in CNNs.

In this work, we propose an alternative way to combine CNNs and GPs, where GPs are approximated using random features expansions (Rahimi and Recht, 2008; Lázaro-Gredilla et al., 2010). The random feature expansion approximation amounts to replacing the orginal kernel matrix with a low-rank approximation, turning GPs into Bayesian linear models. Combining this with CNNs leads to a particular form of Bayesian CNNs, much like GPs and DGPs are particular forms of Bayesian DNNs (Duvenaud et al., 2014; Gal and Ghahramani, 2016a; Neal, 1996). Inference in Bayesian CNNs is intractable and requires some form of approximation. In this work, we draw on the interpretation of dropout as variational inference, employing the so-called Monte Carlo Dropout (MCD, (Gal and Ghahramani, 2016a)) to obtain a practical way of combining CNNs and GPs.

3 State-of-the-art combinations of CNNs and GPs are miscalibrated

Consider a Q-class image classification task where $\mathbf{X}$ denotes a set of n images $\mathbf{x}_i \in \mathbb{R}^{p_x \times p_y} (1 \leq i \leq n)$, and $\mathbf{Y}$ is the matrix consisting of the corresponding one-hot encoded labels $\mathbf{y}_i$ stacked by row. We can use various metrics to determine the quality of a classifier, and here we focus in particular on calibration.

Let $\mathbf{g}(\mathbf{x})$ be the output of a classifier for an input image $\mathbf{x}$. To compute the calibration properties of a classifier, consider a partitioning of the test set $\mathbf{X}_*$ into disjoint sets $\{\mathbf{X}_1, \ldots, \mathbf{X}_M\}$, such that each subset $\mathbf{X}_m$ contains the inputs yielding predictions in the range $(\frac{m-1}{M}, \frac{m}{M}]$. Hence, the confidence associated with each subset $\mathbf{X}_m$ is characterized by the midpoint of its corresponding range, i.e. $\text{conf}(\mathbf{X}_m) = \frac{m-0.5}{M}$. Then, the accuracy $\text{acc}(\mathbf{X}_m)$ for each subset can be evaluated as follows:

$$\frac{1}{|\mathbf{X}_m|} \sum_{\mathbf{x}_* \in \mathbf{X}_m} \delta \left(\arg\max(\mathbf{y}_*) - \arg\max(\mathbf{g}(\mathbf{x}_*)) \right), \quad (1)$$

where $\delta(x)$ is equal to one if $x = 0$, and zero otherwise.

In what follows, we use reliability diagrams to assess calibration, where we plot accuracy as a function of confidence for the subsets $\{\mathbf{X}_1, \ldots, \mathbf{X}_M\}$. For a perfectly calibrated classifier, we expect $\text{acc}(\mathbf{X}_m) = \text{conf}(\mathbf{X}_m)$ for all m, with deviations implying that the class probabilities are either underestimated or overestimated. A useful summary statistics that can be extracted from reliability diagrams is the *Expected Calibration Error* (ECE), which is the average of the absolute difference between accuracy and confidence *weighted* according to its size:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|\mathbf{X}_m|}{|\mathbf{X}_*|} \left| \text{acc}(\mathbf{X}_m) - \text{conf}(\mathbf{X}_m) \right|. \quad (2)$$

Another metric that measures the accuracy in predicting class probabilities is the BRIER score which takes into account the factors of calibration, resolution and uncertainty (Murphy, 1973). It is defined as the squared distance between labels and outputs averaged across classes and test points:

$$\text{BRIER} = \frac{1}{N_{\text{test}}} \sum_{\mathbf{x}_* \in \mathbf{X}_*} \frac{1}{Q} \sum_{k=1}^{Q} \left((\mathbf{y}_*)_k - (\mathbf{g}(\mathbf{x}_*))_k \right)^2. \quad (3)$$

In figure 1, we report the reliability diagrams of three state-of-the-art combinations of CNNs and GPs, i.e GPDNN approach in Bradshaw et al. (2017), CGP in van der Wilk et al. (2017) and SVDKL in Wilson et al. (2016). These approaches are applied to the CIFAR10 and CIFAR100 data sets with various convolutional structures. Note that the lines for CGP in the sub-figure of CIFAR10-LeNet and CIFAR10-RESNET are identical because there is no equivalent CNN architecture in CGP. All of reliability diagrams for these methods and ours can be found in the supplemental material.

The results indicate that current approaches that combine CNNs and GPs are miscalibrated, with a tendence of being overconfident in predictions. This is an important and perhaps surprising finding, because one of the motivations to combine CNNs with GPs is to do better quantification of uncertainty compared to plain CNNs. In the experiments section we report more extensively on the calibration of these classifiers, as well as illustrating other performance metrics. These considerations call for the study of better ways to combine CNNs and GPs to recover calibration while attempting to improve on standard metrics such as error rate and test log-likelihood. The next section illustrates our proposal that achieves this goal.

4 CNN+GP(RF): Conv Nets with Random Feature Expanded GPs

In the proposed model, the labels $\mathbf{Y}_{i\cdot}$ are assumed to be conditionally independent given a set of corresponding latent variables $\mathbf{F}_{i\cdot}$, i.e. we consider the likelihood $p(\mathbf{Y}|\mathbf{F}) = \prod_{i=1}^{n} p(\mathbf{Y}_{i\cdot} | \mathbf{F}_{i\cdot})$, where the latent variables $\mathbf{F}$ are realizations of a set of Q functions $f_j(\mathbf{x})$ at the input images $\mathbf{x}_1, \ldots, \mathbf{x}_n$, i.e., $(\mathbf{F})_{ij} = f_j(\mathbf{x}_i)$ for $j = 1, \ldots, Q$. Each individual $p(\mathbf{Y}_{i\cdot} | \mathbf{F}_{i\cdot})$ is multinomial with probabilities obtained using a softmax transformation of the latent variables. In this work we focus on functions $f_j(\mathbf{x})$ that are modeled using GPs;

note that extension to DGPs is actually easy to consider in our framework, as we show in the experiments.

Due to the GP modeling assumption, the latent function values $\mathbf{F}_{\cdot j} = (f_j(\mathbf{x}_1), \dots, f_j(\mathbf{x}_n))^\top$ are jointly Gaussian with $p(\mathbf{F}_{\cdot j}|\mathbf{X}, \boldsymbol{\theta}) \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$, where $\mathbf{K}$ is the covariance matrix. The entries of the covariance matrix $\mathbf{K} = \{k(\mathbf{x}_i, \mathbf{x}_j|\boldsymbol{\theta})\}_{i,j}$, are specified by a covariance (kernel) function k (with hyperparameters $\boldsymbol{\theta}$) and this form is shared across output dimensions, although this can be relaxed and allow for a different k for the Q outputs.

Instead of applying the GP modeling directly to the images, we propose to employ a transformation $\mathbf{c}(\mathbf{x}|\boldsymbol{\Psi})$ using convolutional layers, where $\boldsymbol{\Psi}$ denotes the parameters of such layers. The vector-valued function $\mathbf{c}(\mathbf{x}|\boldsymbol{\Psi})$ is differentiable as it implements a series of differentiable operations, such as convolutions and pooling. This is one of the key successes of CNN models that allows for the learning of their filters, which we exploit for the end-to-end learning of our model.

Inference in this model requires being able to characterize the posterior over all or a selected group of model parameters, but this posterior is analytically intractable and thus computationally prohibitive (Rasmussen and Williams, 2006). In the remainder of this paper, we build on previous work on scalable inference for GPs and DGPs with random features (Cutajar et al., 2017) to obtain an approximation to the proposed model that can be learned end-to-end.

4.1 Random Feature Expansions

Naïve inference in GP models requires algebraic operations with $\mathbf{K}$ that would cost $\mathcal{O}(n^3)$ in time. Popular approaches to recover tractability use low-rank approximations of the kernel matrix. Among this family of low-rank approximations, we choose to work with random feature approximations (Lázaro-Gredilla et al., 2010; Cutajar et al., 2017). The reason is that they offer a number of possible extensions to speedup computations (e.g., using structured approximations (Le et al., 2013; Yu et al., 2016)) and increase the complexity of the model (e.g., considering Deep GPs (Cutajar et al., 2017)); we elaborate on this in the experiments section. In random feature expansions, the kernel matrix is replaced by a low-rank approximation $\mathbf{K} \approx \boldsymbol{\Phi}\boldsymbol{\Phi}^\top$, with $\boldsymbol{\Phi} \in \mathbb{R}^{n \times m}$ and $m \ll n$. This approximation suggests the construction of a Bayesian linear model to approximate the GP latent variables as $\mathbf{F} = \boldsymbol{\Phi}\mathbf{W}$. Using $p(W_{ij}) = \mathcal{N}(W_{ij}|0, 1)$ it is straightforward to show that the covariance of each of the latent functions $\mathbf{F}_{\cdot j}$ is indeed an approximation to $\mathbf{K}$, as $\mathrm{cov}(\mathbf{F}_{\cdot j}) = \mathrm{E}(\boldsymbol{\Phi}\mathbf{W}_{\cdot j}\mathbf{W}_{\cdot j}^\top\boldsymbol{\Phi}^\top) = \boldsymbol{\Phi}\mathrm{E}(\mathbf{W}_{\cdot j}\mathbf{W}_{\cdot j}^\top)\boldsymbol{\Phi}^\top = \boldsymbol{\Phi}\boldsymbol{\Phi}^\top \approx \mathbf{K}$.

In this work, we focus in particular on the order-one ARC-COSINE kernel (Cho and Saul, 2009)

$$k_{\mathrm{arc}}^{(1)}(\mathbf{x}_i, \mathbf{x}_j|\boldsymbol{\Psi}, \boldsymbol{\theta}) = \frac{\sigma^2}{\pi} \left\|\Lambda^{-\frac{1}{2}}\mathbf{c}(\mathbf{x}_i|\boldsymbol{\Psi})\right\| \left\|\Lambda^{-\frac{1}{2}}\mathbf{c}(\mathbf{x}_j|\boldsymbol{\Psi})\right\| [\sin(\alpha) + (\pi - \alpha)\cos(\alpha)], \quad (4)$$

where $\boldsymbol{\theta} = (\sigma, \boldsymbol{\Lambda} = \mathrm{diag}(\ell_1^2, \dots, \ell_d^2))$ and α is the angle between $\Lambda^{-\frac{1}{2}}\mathbf{c}(\mathbf{x}_i|\boldsymbol{\Psi})$ and $\Lambda^{-\frac{1}{2}}\mathbf{c}(\mathbf{x}_j|\boldsymbol{\Psi})$.

The ARC-COSINE covariance has a convenient integral representation that allows for a Monte Carlo approximation, obtaining a low-rank approximation to the covariance matrix involving Rectified Linear Unit (RELU) activations (Cho and Saul, 2009)

$$\boldsymbol{\Phi}_{\mathrm{arc}} = \sqrt{\frac{2\sigma^2}{N_{\mathrm{RF}}}} \max\left(\mathbf{0}, \mathbf{C}(\mathbf{X}|\boldsymbol{\Psi})\,\boldsymbol{\Omega}\right). \quad (5)$$

In this expression, we have defined $\mathbf{C}(\mathbf{X}|\boldsymbol{\Psi})$ as the matrix resulting from the application of convolutional layers to the image training set $\mathbf{X}$ and $\boldsymbol{\Omega}$ is obtained by stacking N_{RF} samples from $p(\boldsymbol{\omega}) = \mathcal{N}\left(\boldsymbol{\omega}|\mathbf{0}, \boldsymbol{\Lambda}^{-1}\right)$ by column. Note that in the case of a popular Radial Basis Function (RBF) covariance, it is possible to obtain a similar random feature approximation, where the RELU activation is replaced by trigonometric functions; see Rahimi and Recht (2008) and the supplement for details.

4.2 End-to-end learning

Inference in the proposed model is intractable due to the likelihood that is not conjugate to the GP prior. Further complications stem from the need to infer kernel parameters, which include convolutional parameters, and the need to be able to scale to large data. Our aim is to carry out inference within a consistent framework that is characterized by simplicity, as described next.

We start by introducing an approximate posterior over $\mathbf{W}, \boldsymbol{\Omega}$ and $\boldsymbol{\Psi}$, that we denote as $q(\mathbf{W}, \boldsymbol{\Omega}, \boldsymbol{\Psi})$. Following standard variational inference arguments, we can define an operative way to obtain these approximate posteriors. The log-marginal likelihood $\mathcal{L} = \log[p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}]$ can be bounded by the sum of an expected log-likelihood term and a negative Kullback-Leibler (KL) divergence term as follows:

$$\mathcal{L} \geq \mathrm{E}_{q(\mathbf{W}, \boldsymbol{\Omega}, \boldsymbol{\Psi})}\left(\log\left[p\left(\mathbf{Y}|\mathbf{X}, \mathbf{W}, \boldsymbol{\Omega}, \boldsymbol{\Psi}, \boldsymbol{\theta}\right)\right]\right) - \mathrm{KL}\left[q\left(\mathbf{W}, \boldsymbol{\Omega}, \boldsymbol{\Psi}\right) \| p\left(\mathbf{W}, \boldsymbol{\Omega}, \boldsymbol{\Psi}\right)\right]. \quad (6)$$

Variational inference amounts to optimizing the lower bound above with respect to $q(\mathbf{W}, \boldsymbol{\Omega}, \boldsymbol{\Psi})$ and any other parameters of interest.

We have now a number of options on the form for the approximate posteriors $q(\mathbf{W}, \boldsymbol{\Omega}, \boldsymbol{\Psi})$. In previous works on

variational inference for DNNs, it has been proposed to define the approximating distributions to be Gaussian and factorized across parameters (Kingma and Welling, 2014; Graves, 2011). The drawback of this is that it doubles the number of parameters. Alternatively, we can rely on the connections between dropout and variational inference (Gal and Ghahramani, 2016a,b) which is drawn by assuming the posterior of $\mathbf{W}, \mathbf{\Omega}$ and $\mathbf{\Psi}$ as a mixture of two Gaussian distributions (see supplement). From this connection, we are able to obtain an easier approximate inference scheme, which is also known as Monte Carlo Dropout (MCD). Focusing on the weights for now, the connection with dropout is apparent if we rewrite

$$\mathbf{W} = \mathbf{M}_w \operatorname{diag}[\mathbf{z}_w] \tag{7}$$

with $(\mathbf{z}_w)_i \sim \text{Bernoulli}(\pi_w)$. The reparameterization introduces variational parameters $\mathbf{M}_w$ (one for each weight in $\mathbf{W}$) and a vector of binary variables that can switch on or off the columns of the weight matrix with probability π_w. A similar reprameterization can be done for the convolutional parameters $\mathbf{\Psi}$ and matrices of random feature $\mathbf{\Omega}$, introducing $\mathbf{M}_\psi, \mathbf{M}_\Omega$ and π_ψ, π_Ω. The optimization of the lower bound wrt all variational parameters requires being able to evaluate the expectation and the KL term in (6).

In MCD, the KL term in (6) can be approximated following Gal and Ghahramani (2016a), obtaining a regularization term involving the squared-norm of the parameters

$$\begin{aligned}\mathrm{KL}\left[q\left(\mathbf{W}, \mathbf{\Omega}, \mathbf{\Psi}\right) \| p\left(\mathbf{W}, \mathbf{\Omega}, \mathbf{\Psi}\right)\right] \approx \\ \frac{\pi_w}{2}\left\|\mathbf{M}_w\right\|^2 + \frac{\pi_\Omega}{2}\left\|\mathbf{M}_\Omega\right\|^2 + \frac{\pi_\psi}{2}\left\|\mathbf{M}_\psi\right\|^2\end{aligned} \tag{8}$$

The expectation in (6), instead, can be unbiasedly estimated using Monte Carlo and also considering a mini-batch of size m:

$$\frac{n}{m}\frac{1}{N_{\mathrm{MC}}}\sum_{i=1}^{N_{\mathrm{MC}}}\sum_{k\in\mathcal{I}_m}\log\left[p\left(\mathbf{y}_k|\mathbf{x}_k, \mathbf{W}^{(i)}, \mathbf{\Omega}^{(i)}, \mathbf{\Psi}^{(i)}, \boldsymbol{\theta}\right)\right]$$

with $\mathbf{W}^{(i)}, \mathbf{\Omega}^{(i)}, \mathbf{\Psi}^{(i)} \sim q(\mathbf{W}, \mathbf{\Omega}, \mathbf{\Psi})$, and $\mathcal{I}_m$ is a set of m indices to select a mini-batch of training points (Graves, 2011). This doubly-stochastic approximation is differentiable wrt variational parameters when the Bernoulli variables are fixed.

The approximate objective can now be optimized in the same vein as in standard back-propagation with dropout, noting that dropout is applied to $\mathbf{W}$, $\mathbf{\Omega}$ and to convolutional parameters $\mathbf{\Psi}$. What changes, however, is the interpretation of the procedure as stochastic variational inference, whereby the Bernoulli variables are resampled at each iteration. A practical implication is in the way we compute the predictive distribution, which has a probabilistic flavor as follows:

$$\begin{aligned}p(\mathbf{y}_*|\mathbf{x}_*, X, \boldsymbol{\theta}) \approx \int p(\mathbf{y}_*|\mathbf{W}, \mathbf{\Omega}, \mathbf{\Psi}, \mathbf{x}_*, X, \boldsymbol{\theta}) \\ q(\mathbf{W}, \mathbf{\Omega}, \mathbf{\Psi}) d\mathbf{W} d\mathbf{\Omega} d\mathbf{\Psi},\end{aligned} \tag{9}$$

and can be approximated using Monte Carlo by resampling the Bernoulli variables. While MCD has been proposed for CNNs in (Gal and Ghahramani, 2016b), in this work we extend it to the case of joint inference over convolutional parameters and the GP approximation in the CNN+GP(RF) model, thus obtaining a practical inference and prediction scheme, which combines CNNs and GPs.

4.3 Extensions

Structured random feature approximations: One of the advantages of the proposed model, compared to other GP approximations, is that it can exploit structured random feature expansions to accelerate computations and reduce the size of the approximate GP (Le et al., 2013; Yu et al., 2016). In the random features approximation, random features are constructed by multiplying $\mathbf{\Omega}$ with the convolutional features. Without loss of generality, assuming that $\mathbf{\Omega} \in \mathbb{R}^{m\times d}$ and $\mathbf{c}(\mathbf{x}|\mathbf{\Psi}) \in \mathbb{R}^{d\times 1}$, the cost of computing products $\mathbf{\Omega}\mathbf{c}(\mathbf{x}|\mathbf{\Psi})$ is $\mathcal{O}(md)$, while storing $\mathbf{\Omega}$ requires $\mathcal{O}(md)$ storage.

Structured approximations aim to reduce the time complexity to $\mathcal{O}(m \log d)$ and the storage cost to $\mathcal{O}(m + d)$. Taking a standard random features expansion of the isotropic covariance in (5) with $\mathbf{\Lambda} = \ell^{-2}\mathbf{I}$ as an example, $\mathbf{\Omega} = \frac{1}{\ell}\mathbf{G}$, with $\mathbf{G}_{ij} \sim \mathcal{N}(0, 1)$. One way to make computations cheaper is to replace the Gaussian matrix $\mathbf{G}$ with a pseudo-random alternative. The Structured Orthogonal Random Feature (SORF) approximation (Yu et al., 2016) approximates $\mathbf{G}$ through a series of Hadamard transformations of diagonal matrices $\mathbf{D}_i$ with elements randomly sampled from $\{-1, +1\}$ or Rademacher distribution, that is $\mathbf{G} \approx \sqrt{d}\mathbf{H}\mathbf{D}_1\mathbf{H}\mathbf{D}_2\mathbf{H}\mathbf{D}_3$, where $\mathbf{H}$ is the normalized Walsh-Hadamard matrix. We refer to this variation of the model as CNN+GP(SORF).

Similarly to the other parameters, we infer the diagonal matrices $\mathbf{D}_i$ using MCD. We denote by $\mathbf{d}_i$ the diagonal of $\mathbf{D}_i, i = 1, 2, 3$. The MCD scheme (Gal and Ghahramani, 2016a,b) assumes an L_2 regularization which implies a zero-mean Gaussian prior, which is inappropriate for $\mathbf{d}_i$ as it is Rademacher distributed. We propose to bypass this limitation by applying MCD to a reparameterization of $\mathbf{d}_i$. In particular, denoting by $\mathbf{d}_i^* \in \{-1, +1\}^d$ the initialized values of $\mathbf{d}_i$, we apply MCD to $\mathbf{d}_i - \mathbf{d}_i^*$. According to this choice, each diagonal element is sampled based on the variational

Table 1: CNN architectures considered in this work. The same architectures are used in GPDNN and SVDKL by replacing the fully connected layers with GPs, while CGP does not explicitly use a convolutional structure.

Depth	Data set	CNN architecture	CNN name	# Conv features
Shallow	MNIST	2 Conv Layers + 2 Fully connected	LeNet	4096
Shallow	CIFAR10	2 Conv Layers + 3 Fully connected	LeNet	4096
Deep	CIFAR10	30 Conv Layers + 1 Fully connected	RESNET	64
Deep	CIFAR100	150 Conv Layers + 1 Fully connected	RESNET	64

parameters $\mathbf{M}_{\mathbf{d}_i - \mathbf{d}_i^*}$

$$\mathbf{d}_i = \begin{cases} \mathbf{M}_{\mathbf{d}_i - \mathbf{d}_i^*} + \mathbf{d}_i^*, & \text{with probability } \pi_d \\ \mathbf{d}_i^*, & \text{otherwise} \end{cases} \quad (10)$$

Convolutional Networks with Random-Feature-Expanded Deep GPs: A DGP model represents a deep probabilistic nonparametric approach where the output of one GP at each layer is used as the input to the GP in the next layer (Damianou and Lawrence, 2013). Extending the random feature approximation to DGPs and the inference scheme presented here is straightforward; see Cutajar et al. (2017) for details. The random feature approximation turns the DGP into a Bayesian DNN for which we can apply stochastic variational inference to infer model parameters. In the experiments section, we explore the possibility to stack a DGP on top of convolutional layers, and we show the impact of depth on performance.

5 Experiments

We carry out the experimental evaluation using popular benchmark datasets, such as MNIST, CIFAR10 and CIFAR100 and with a number of popular CNN architectures based on LeNet and RESNET (see table 1).

We report three state-of-the-art competitors combining CNNs and GPs, namely GPDNN (Bradshaw et al., 2017), SVDKL (Wilson et al., 2016), and CGP (van der Wilk et al., 2017). We also report Bayesian CNNs, as suggested in Gal and Ghahramani (2016b) and CNNs with post-calibration as proposed in Guo et al. (2017), which we refer to as CNN+MCD and CNN+CAL, respectively. For all the competing methods we used available implementations, adding the same CNN architecture to ensure a fair comparison. In all experiments, we use a batch-size $m = 100$ and the Adam optimizer with default learning rate (Kingma and Ba, 2017). In the methods that use MCD, we use a dropout rate of 0.5 for all parameters.

The results are reported in figure 2, where we have used different training sizes n, keeping the classes balanced. In the figure, we report the calibration measures that we have introduced earlier, namely ECE and BRIER scores, and we also report the classification error rate (ERR) and the mean negative test log-likelihood (MNLL). Compared to other combinations of CNNs and GPs, CNN+GP(RF) improves considerably on all metrics. It is interesting to see that our proposal is competitive with Bayesian CNNs employing MCD, with only a marginal improvement on ERR and MNLL in some configurations.

In CAL it is necessary to leave out part of the data to perform post-calibration, which can be problematic in applications where obtaining labeled data is difficult or expensive. As a result, our proposal is considerably better, although CAL is competitive in ECE; this is expected given that this is the metric that is optimized after training.

The two variants of our approach, namely CNN+GP(RF) where we learn the frequencies $\boldsymbol{\Omega}$ and CNN+GP(SORF) where we sample $\boldsymbol{\Omega}$ from its prior, are comparable. This suggests that the extra level of complexity of learning the spectral frequencies does not lead to substantial gains in performance and that the structured random feature approximation yields satisfactory performance.

We also note that these results have been obtained by fixing the covariance parameters $\boldsymbol{\theta}$ of the GP, as we found it to be unstable when learning these jointly with $\boldsymbol{\Omega}$. This might be the reason why these parameters were learned through cross-validation in Gal et al. (2017). In the supplement, we report the results obtained when learning $\boldsymbol{\theta}$ and fixing $\boldsymbol{\Omega}$, which we found yielding similar performance as fixing $\boldsymbol{\theta}$. All these observations corroborate the hypothesis that most of the performance of CNN-based classification models is due to the convolutional layers.

In summary, figure 2 shows that our CNN+GP(RF) is the best strategy for calibrating these models compared to other approaches using GPs. Furthermore, we found perhaps surprisingly that MCD has comparable performance. In the supplementary material, we report results on GPDNN where we infer convolutional parameters using MCD, so as to gain insights as to whether most of the improvements in performance are due to this form of regularization. The results support the intuition that inferring these parameters yields improve-

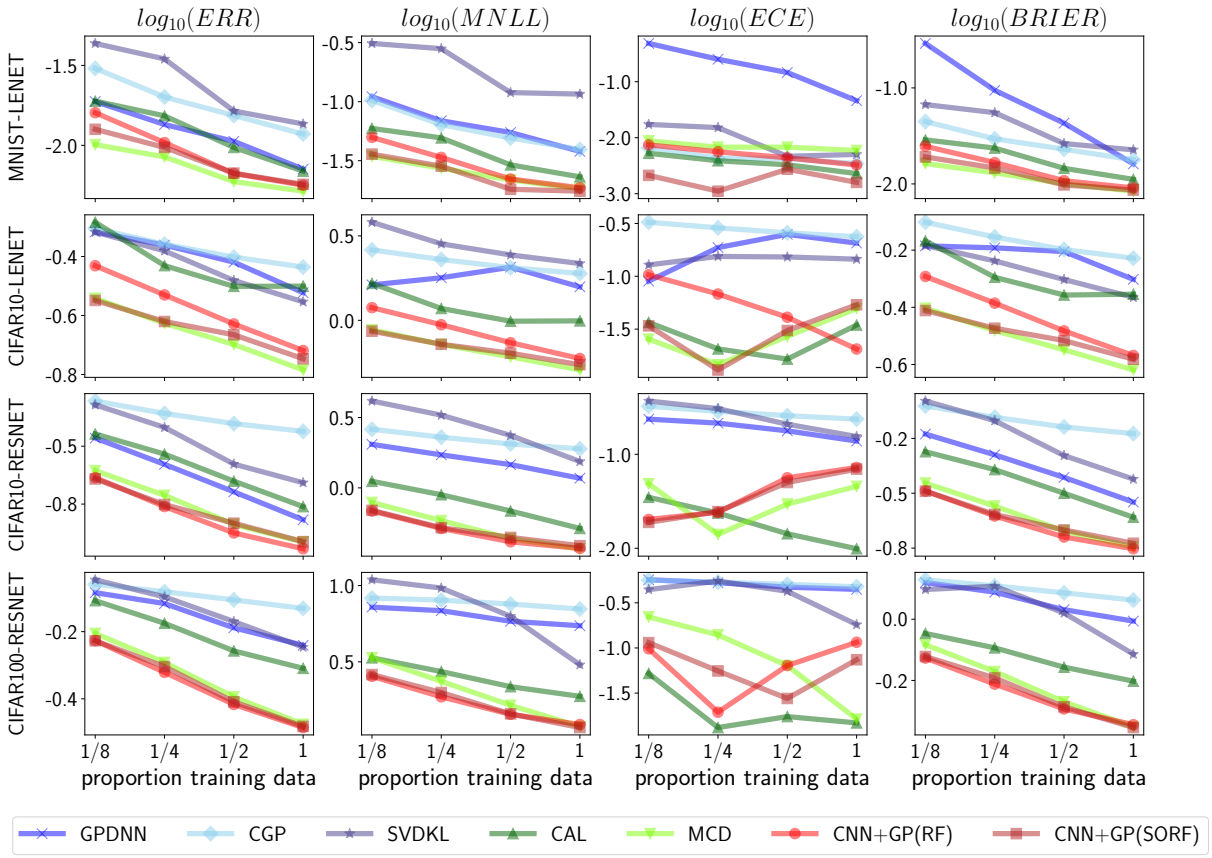

Figure 2: Comparison of our CNN+GP(RF) and CNN+GP(SORF) with existing combinations of CNNs with GPs, and with Bayesian CNNs and post-calibrated CNNs. All performace metrics are defined so that the lower the better.

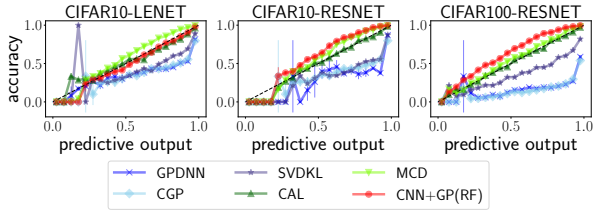

Figure 3: Reliability diagrams of our CNN+GP(RF) in comparison with existing combinations of CNNs with GPs, and with Bayesian CNNs and post-calibrated CNNs.

ments in calibration, but also that our CNN+GP(RF) still offers better performance.

Reliability diagrams: In figure 3, we report the reliability diagrams of all the methods studied in figure 1. The figure shows that CAL, MCD and CNN+GP(RF) produce well-calibrated predictions when using a shallow convolutional structure (LENET). For a deeper architecture (RESNET), CNN+GP(RF) is slightly underconfident. Compared to previous combinations of CNNs and GPs, our approach yields better reliability curves.

Experiments combining CNNs and Deep GPs: In figure 4, we report results varying the depth of a DGP on top of the convolutional layers; again, we learn the convolutional filters and the DGP end-to-end as discussed in the previous sections. We show results when applying our model to the whole CIFAR10 data set in the case of the shallow convolutional structure (table 1). We feed-forward the convolutional features to all layers of the DGP, in line with what suggested in the literature of DGPs to avoid pathologies in the functions that can be modeled (Cutajar et al., 2017; Duvenaud et al., 2014; Neal, 1996). The results indicate that increasing the complexity of the model improves on all performance metrics, and worsen calibration, which however is still around 3% ECE. This is in line with the intuition that increasing model complexity negatively impacts calibration.

Knowing when the model doesn't know: We report experiments showing the ability of our model to know when it does not know, following a similar experimental setup as in Lakshminarayanan et al. (2017). In this experiment we train our CNN+GP(RF) model on MNIST and test on the NOT-MNIST dataset, which

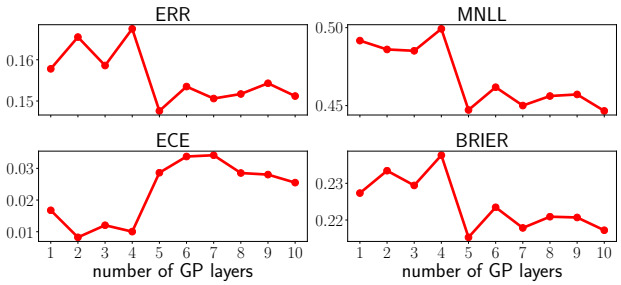

Figure 4: Performance of the proposed model when varying the depth of the DGP on top of a RESNET convolutional structure on CIFAR10 dataset. Note that the scale of y-axes indicates that the metrics change only slightly when increasing the depth of the DGP.

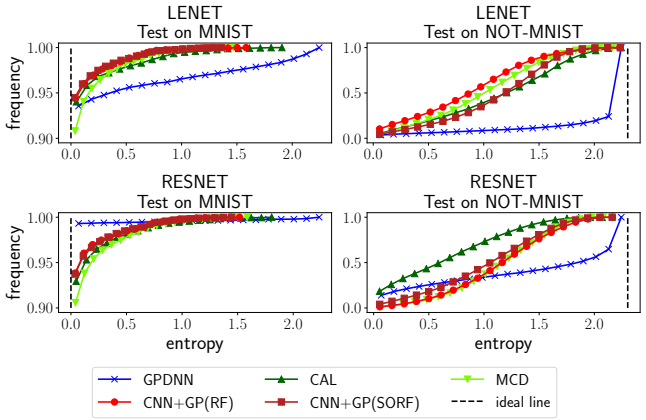

Figure 5: Cumulative distribution function plot of predictive entropies when the models trained on MNIST are tested on MNIST and NOT-MNIST. We report results for two different depths of the convolutional structure. NOT-MNIST dataset available at `http://yaroslavvb.blogspot.fr/2011/09/notmnist-dataset.html`

contains images of letters from "A" to "J" in various typefaces. For this experiment, while we do not know the exact value that we should obtain for predictive probabilities, we expect to observe low entropy in the predictions when tesing on MNIST and high entropy when predicting on NOT-MNIST, indicating high uncertainty. The results are reported in figure 5, where we show the cumulative distribution of the entropy of predictive probabilities for two depths of the convolutional structure. In the figure, we compare our CNN+GP(RF) against one of the methods combining CNNs and GPs, that is GPDNN. In the figure, we also include results on CNNs with post-calibration and Bayesian CNNs inferred with MCD. Our approach is competitive with Bayesian CNNs and it is considerably superior to post-calibration. This is especially true in the case of the RESNET convolutional structure, where post-calibration still yields a large number of predictions with low uncertainty. Interestingly, GPDNN assigns large uncertainty to predictions on NOT-MNIST, although with the deeper convolutional architecture it yields a large fraction of predictions with low entropy. We speculate that this due to the inducing point approximation of the GP, which nicely captures uncertainty away from training data except for test points which are closer to the training data.

Replacing fully connected layers by GP with the SORF approximation: In table 2, we report further results comparing MCD with CNN+GP(SORF). In this experiment, we use the ALEXNET structure (Krizhevsky et al., 2012) on CIFAR10 and CIFAR100 datasets. The results in table 2 show improvements in using our model compared CNNs with MCD. We attribute this to the fact that the GP approximated through SORF in place of the fully connected layer of ALEXNET reduces model parameters from 30 million to 2.3 million.

Table 2: Comparison between CNN+GP(SORF) and MCD with ALEXNET architecture.

METHOD	ERR	MNLL	ECE	BRIER
CNN+GP(SORF)	0.172	0.522	0.063	0.250
MCD	0.181	0.591	0.110	0.276

(a) CIFAR10

METHOD	ERR	MNLL	ECE	BRIER
CNN+GP(SORF)	0.459	1.806	0.127	0.612
MCD	0.594	2.434	0.058	0.732

(b) CIFAR100

6 Conclusions

Despite the considerable interest in combining CNNs with GPs, little attention has been devoted to understand the implications in terms of the ability of these models to accurately quantify the level of uncertainty in predictions. This is the first work that highlights the issues of calibration of these models, showing that GPs cannot cure the issues of miscalibration in CNNs. We have proposed a novel combination of CNNs and GPs where the resulting model becomes a particular form of a Bayesian CNN for which inference using variational inference is straightforward. However, our results also indicate that combining CNNs and GPs does not generally improve the performance of standard CNNs. This can serve as a motivation for investigating new approximation methods for scalable inference in GP models and combinations with CNNs.

Acknowledgements

JPC acknowledges support from the Simons Foundation and the McKnight Foundation. MF gratefully acknowledges support from the AXA Research Fund.

References

J. Bradshaw, Alexander, and Z. Ghahramani. Adversarial Examples, Uncertainty, and Transfer Testing Robustness in Gaussian Process Hybrid Deep Networks, July 2017. arXiv:1707.02476.

R. Calandra, J. Peters, C. E. Rasmussen, and M. P. Deisenroth. Manifold Gaussian Processes for regression. In *2016 International Joint Conference on Neural Networks, IJCNN 2016, Vancouver, BC, Canada, July 24-29, 2016*, pages 3338–3345, 2016.

Y. Cho and L. K. Saul. Kernel Methods for Deep Learning. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 342–350. Curran Associates, Inc., 2009.

K. Cutajar, E. V. Bonilla, P. Michiardi, and M. Filippone. Random feature expansions for deep Gaussian processes. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 884–893, International Convention Centre, Sydney, Australia, Aug. 2017. PMLR.

A. C. Damianou and N. D. Lawrence. Deep Gaussian Processes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2013, Scottsdale, AZ, USA, April 29 - May 1, 2013*, volume 31 of *JMLR Proceedings*, pages 207–215. JMLR.org, 2013.

D. K. Duvenaud, O. Rippel, R. P. Adams, and Z. Ghahramani. Avoiding pathologies in very deep networks. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, volume 33 of *JMLR Workshop and Conference Proceedings*, pages 202–210. JMLR.org, 2014.

P. A. Flach. Classifier Calibration. In C. Sammut and G. I. Webb, editors, *Encyclopedia of Machine Learning and Data Mining*, pages 1–8. Springer US, Boston, MA, 2016.

Y. Gal and Z. Ghahramani. Dropout As a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1050–1059. JMLR.org, 2016a.

Y. Gal and Z. Ghahramani. Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference, Jan. 2016b. arXiv:1506.02158.

Y. Gal and R. Turner. Improving the Gaussian Process Sparse Spectrum Approximation by Representing Uncertainty in Frequency Inputs. In F. R. Bach and D. M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 655–664. JMLR.org, 2015.

Y. Gal, J. Hron, and A. Kendall. Concrete Dropout. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3581–3590. Curran Associates, Inc., 2017.

A. Graves. Practical Variational Inference for Neural Networks. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2348–2356. Curran Associates, Inc., 2011.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On Calibration of Modern Neural Networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330, International Convention Centre, Sydney, Australia, Aug. 2017. PMLR.

A. Kendall and Y. Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5574–5584. Curran Associates, Inc., 2017.

D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization, Jan. 2017. arXiv:1412.6980.

D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *Proceedings of the Second International Conference on Learning Representations (ICLR 2014)*, Apr. 2014.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

M. Kull, T. S. Filho, and P. Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In A. Singh and J. Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine*

Learning Research, pages 623–631, Fort Lauderdale, FL, USA, Apr. 2017. PMLR.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6402–6413. Curran Associates, Inc., 2017.

M. Lázaro-Gredilla, J. Quinonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal. Sparse Spectrum Gaussian Process Regression. *Journal of Machine Learning Research*, 11:1865–1881, 2010.

Q. Le, T. Sarlos, and A. Smola. Fastfood - Approximating Kernel Expansions in Loglinear Time. In *30th International Conference on Machine Learning (ICML)*, 2013.

D. J. C. Mackay. Bayesian methods for backpropagation networks. In E. Domany, J. L. van Hemmen, and K. Schulten, editors, *Models of Neural Networks III*, chapter 6, pages 211–254. Springer, 1994.

A. H. Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology*, 12(4):595–600, 1973.

R. M. Neal. *Bayesian Learning for Neural Networks (Lecture Notes in Statistics)*. Springer, 1 edition, Aug. 1996. ISBN 0387947248.

A. Niculescu-Mizil and R. Caruana. Predicting Good Probabilities with Supervised Learning. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 625–632, New York, NY, USA, 2005. ACM.

J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10 (3), 1999.

A. Rahimi and B. Recht. Random Features for Large-Scale Kernel Machines. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1177–1184. Curran Associates, Inc., 2008.

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

M. van der Wilk, C. E. Rasmussen, and J. Hensman. Convolutional Gaussian Processes, Sept. 2017.

A. G. Wilson, Z. Hu, R. R. Salakhutdinov, and E. P. Xing. Stochastic Variational Deep Kernel Learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2586–2594. Curran Associates, Inc., 2016.

F. X. Yu, A. T. Suresh, K. M. Choromanski, D. N. Holtmann-Rice, and S. Kumar. Orthogonal Random Features. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1975–1983. Curran Associates, Inc., 2016.

B. Zadrozny and C. Elkan. Transforming Classifier Scores into Accurate Multiclass Probability Estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 694–699, New York, NY, USA, 2002. ACM.