The Promises and Pitfalls of Deep Kernel Learning

Sebastian W. Ober[1] **Carl E. Rasmussen**[1,2] **Mark van der Wilk**[3]

[1]Department of Engineering, University of Cambridge, Cambridge, United Kingdom
[2]Secondmind.ai, Cambridge, United Kingdom
[3]Department of Computing, Imperial College London, London, United Kingdom

Abstract

Deep kernel learning and related techniques promise to combine the representational power of neural networks with the reliable uncertainty estimates of Gaussian processes. One crucial aspect of these models is an expectation that, because they are treated as Gaussian process models optimized using the marginal likelihood, they are protected from overfitting. However, we identify pathological behavior, including overfitting, on a simple toy example. We explore this pathology, explaining its origins and considering how it applies to real datasets. Through careful experimentation on UCI datasets, CIFAR-10, and the UTKFace dataset, we find that the overfitting from overparameterized deep kernel learning, in which the model is "somewhat Bayesian", can in certain scenarios be worse than that from not being Bayesian at all. However, we find that a fully Bayesian treatment of deep kernel learning can rectify this overfitting and obtain the desired performance improvements over standard neural networks and Gaussian processes.

1 INTRODUCTION

Gaussian process (GP) models [Rasmussen and Williams, 2006] are popular choices for Bayesian modeling due to their interpretable nature and reliable uncertainty estimates. These models typically involve only a handful of kernel hyperparameters, which are optimized with respect to the marginal likelihood in an empirical Bayes, or type-II maximum likelihood, approach. However, for most popular choices the kernel itself is fixed, meaning that GP models are unable to learn representations from the data that might aid predictions, and instead act mostly as smoothing devices. This greatly limits the applicability of GPs to high-dimensional and highly structured data, such as images.

Deep neural networks [LeCun et al., 2015], on the other hand, are known to learn powerful representations which are then used to make predictions on unseen test inputs. While deterministic neural networks have achieved state-of-the-art performance throughout supervised learning and beyond, they suffer from overconfident predictions [Guo et al., 2017], and do not provide reliable uncertainty estimates. The Bayesian treatment of neural networks attempts to address these issues; however, despite recent advances in variational inference (e.g.Ober and Aitchison [2020], Dusenberry et al. [2020]) and sampling methods (e.g. Heek and Kalchbrenner [2019], Zhang et al. [2020]) for Bayesian neural networks (BNNs), inference in BNNs remains difficult due to complex underlying posteriors and the large number of parameters in modern BNNs. Moreover, BNNs generally require multiple forward passes to obtain multiple samples of the predictive posterior to average over.

It is natural, therefore, to try to combine the uncertainty-representation advantages of Gaussian processes with the representation-learning advantages of neural networks, and thus obtain the "best of both worlds." Indeed, many works have attempted to achieve this. In this paper, we focus on a line of work that we refer to broadly as *deep kernel learning* (DKL) [Calandra et al., 2016, Wilson et al., 2016a,b]. These works use a neural network to map inputs to points in an intermediate feature space, which is then used as the input space for a Gaussian process. The network parameters can be treated as hyperparameters of the kernel, and thus are optimized with respect to the (log) marginal likelihood, as in standard GP inference. This leads to an end-to-end training scheme that results in a model that hopefully benefits from the representational power of neural networks while also enjoying the benefits of reliable uncertainty estimation from the GP. Moreover, as the feature extraction done by the neural network is deterministic, inference only requires one forward pass of the neural net, unlike fully Bayesian BNNs. Previous works have shown that these methods can be used successfully [Calandra et al., 2016, Wilson et al., 2016a,b, Bradshaw et al., 2017].

Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI 2021), PMLR 161:1206–1216.

We investigate to what extent DKL is actually able to achieve flexibility and good uncertainty, and what makes it successful in practice: for DKL to be useful from a Bayesian perspective, a higher marginal likelihood should lead to better performance. In particular, it is often claimed that optimizing the marginal likelihood will automatically calibrate the complexity of the model, preventing overfitting. For instance, Wilson et al. [2016a] states "the information capacity of our model grows with the amount of available data, but its complexity is automatically calibrated through the marginal likelihood of the Gaussian process, without the need for regularization or cross-validation." This claim is based on the common decomposition of the log marginal likelihood into "data fit" and "complexity penalty" terms [Rasmussen and Williams, 2006], which leads to the belief that a better marginal likelihood will result in better test performance.

This is generally true when selecting a small number of hyperparameters. However, in models like DKL which introduce many hyperparameters, we show that in some cases, marginal likelihood training can encourage overfitting that is *worse* than that from a standard, deterministic neural network. This is because the marginal likelihood tries to correlate *all* the datapoints, rather than just those for which correlations will be important. As most standard Gaussian process models typically only have a few hyperparameters, this sort of overfitting is not usually an issue, but when many hyperparameters are involved, as in DKL, they can give the model the flexibility to overfit in this way. As such, our work has implications for all GP methods which use highly parameterized kernels, as well as methods that optimize more than a handful of model parameters according to the marginal likelihood or ELBO.

In this work, we make the following claims:

- Using the marginal likelihood can lead to overfitting for DKL models.
- This overfitting can actually be worse than the overfitting observed using standard maximum likelihood approaches for neural networks.
- The marginal likelihood overfits by overcorrelating the datapoints, as it tries to correlate all the data, not just the points that should be correlated.
- Stochastic minibatching can mitigate this overfitting, and helps to explain the success of DKL models in practice.
- A fully Bayesian treatment of deep kernel learning can avoid overfitting and obtain the benefits of both neural networks and Gaussian processes.

We note that some works have discussed that overfitting can be an issue for Gaussian processes trained with the marginal likelihood [Rasmussen and Williams, 2006, Cawley and Talbot, 2010, Lalchand and Rasmussen, 2020], and Calandra et al. [2016] mentions that overfitting can be an issue for their DKL model. However, we are not aware of any work that tries to understand the pathological behavior that DKL methods can exhibit or the mechanism with which the marginal likelihood overfits.

2 RELATED WORK

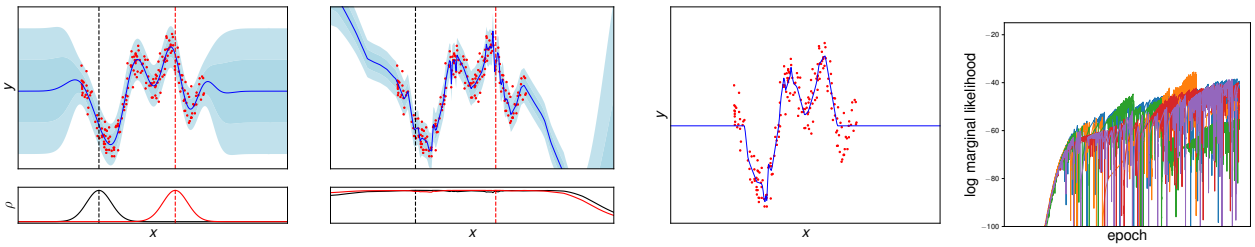
Salakhutdinov and Hinton [2007] used deep belief networks to pretrain a neural network feature extractor to transform the inputs to a GP, with subsequent fine-tuning using the marginal likelihood. Calandra et al. [2016] removed the deep belief network pretraining and only used the marginal likelihood to train the model, referring to this as the "manifold GP". Wilson et al. [2016a] improved the scalability of this model by using KISS-GP [Wilson and Nickisch, 2015], coining the term "deep kernel learning". This was further extended to non-regression likelihoods and multiple outputs in Wilson et al. [2016b] by using stochastic variational inference [Hensman et al., 2015] and Kronecker and Toeplitz structure [Wilson et al., 2015], resulting in stochastic variational deep kernel learning (SVDKL). These and related approaches which use the marginal likelihood to optimize the neural network parameters have been shown to be advantageous in multiple situations, including transfer testing and adversarial robustness [Bradshaw et al., 2017]. On the other hand, Tran et al. [2019] investigated poor calibration in these models, and proposed using Monte Carlo dropout [Gal and Ghahramani, 2016] to perform approximate Bayesian inference over the neural network weights in the model. However, they did not explain the poor calibration, nor did they identify the possibility of overfitting in DKL models.

Due to the difficulty of performing full inference over all BNN parameters, there has been a recent increase in interest in using deterministic feature extractors for models that only incorporate uncertainty in an output layer (e.g. Liu et al. [2020], van Amersfoort et al. [2020]). One of the most popular models in recent years has been the "neural linear" model [Riquelme et al., 2018, Ober and Rasmussen, 2019], which can be viewed as DKL with a linear kernel, or equivalently, Bayesian inference over the last layer of a neural network. In particular, Ober and Rasmussen [2019] showed that it is difficult to get the neural linear model to perform well for regression without considerable hyperparameter tuning, and that fully Bayesian approaches for BNNs often require much less tuning to obtain comparable results. Recent approaches (e.g. Liu et al. [2020], van Amersfoort et al. [2021]) try to regularize the neural network to mitigate these issues, but in doing so introduce additional hyperparameters that require tuning.

3 BACKGROUND

3.1 GAUSSIAN PROCESSES

A Gaussian process is a collection of random variables such that every finite collection of these random variables is



(a) SE kernel (b) Exact DKL kernel (c) Neural network fit (d) LML training curves

Figure 1: Results on the toy dataset from Snelson and Ghahramani [2006]. (a) and (b) show plots of the predictive posterior for squared exponential (SE) and deep kernel learning (DKL) kernels, respectively; below each plot we also plot correlation functions $\rho_{x'}(x) = k(x, x')/\sigma_f^2$ at two points x' given by the vertical dashed lines. (c) shows the fit given by the neural network analogous to the DKL model. Finally, (d) shows training curves of the log marginal likelihood (LML) for 5 different initializations of DKL.

distributed according to a multivariate normal distribution. In the regression setting, where we have a dataset consisting of inputs $X = (x_1, \ldots, x_N)^T$, $x_n \in \mathbb{R}^D$, and outputs $\mathbf{y} = (y_1, \ldots, y_N)^T$, $y_n \in \mathbb{R}$, we assume that each datapoint is generated according to

$$y_n = f(x_n) + \epsilon_n, \ \epsilon_n \sim \mathcal{N}(0, \sigma_n^2), \tag{1}$$

where f is drawn from a Gaussian process prior, $f \sim \mathcal{GP}(m, k)$. Here, $m : \mathbb{R}^D \to \mathbb{R}$ is the mean function, and $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is a symmetric, positive semi-definite covariance (kernel) function. Together, these uniquely define the Gaussian process prior: for instance, the marginal distribution indexed by X is distributed according to $\mathcal{N}(\mathbf{m}(X), K)$, where $\mathbf{m}(X) = (m(x_1), \ldots, m(x_n))^T$ and we define the kernel matrix $K \coloneqq K(X, X)$, so that $K_{ij} = k(x_i, x_j)$.

Predictions of the latent function for a collection of test points X_* can be computed in closed form:

$$\mathbf{f}_*|X, \mathbf{y}, X_* \sim \mathcal{N}(\mu_*, \Sigma_*), \text{ where} \tag{2}$$
$$\mu_* = \mathbf{m}(X_*) + K(X_*, X)(K + \sigma_n^2 I_N)^{-1}(\mathbf{y} - \mathbf{m}(X)),$$
$$\Sigma_* = K(X_*, X_*) - K(X_*, X)(K + \sigma_n^2 I_N)^{-1}K(X, X_*).$$

For the purposes of this work, we take the mean function to be zero.

Finally, it is typical for the kernel to have a number of hyperparameters which are learned along with the noise variance, σ_n^2, by maximizing the (log) marginal likelihood (LML; also known as the model evidence), in an empirical Bayes, or type-II maximum likelihood approach:

$$\log p(\mathbf{y}) = \log \mathcal{N}(\mathbf{y}|\mathbf{0}, K + \sigma_n^2 I_N) \tag{3}$$
$$\stackrel{c}{=} \underbrace{-\frac{1}{2}\log|K + \sigma_n^2 I_N|}_{\text{(a) complexity}} \underbrace{-\frac{1}{2}\mathbf{y}^T(K + \sigma_n^2 I_N)^{-1}\mathbf{y}}_{\text{(b) data fit}},$$

where we note that (a) and (b) are often referred to as the "complexity penalty" and "data fit" terms, respectively. For the purposes of this work, we use the automatic relevance determination (ARD) squared-exponential (SE) kernel, $k(x, x') = \sigma_f^2 \exp(-\frac{1}{2}\sum_{d=1}^{D}(x_d - x'_d)^2/l_d^2)$. Therefore, the hyperparameters to tune are the noise variance, σ_n^2, signal variance, σ_f^2, and lengthscales l_d^2.

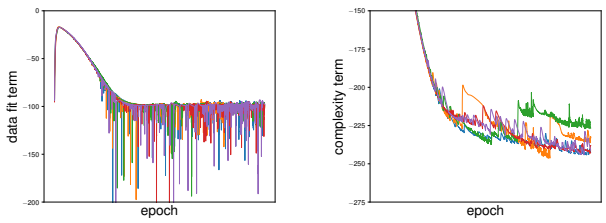
3.2 DEEP KERNEL LEARNING

One of the central critiques of Gaussian process regression is that it does not actually learn representations of the data. In an attempt to address this, several works [Calandra et al., 2016, Wilson et al., 2016a,b, Bradshaw et al., 2017] have proposed variants of deep kernel learning (DKL), which maps the inputs x_n to intermediate values $v_n \in \mathbb{R}^Q$ through a neural network $g_\phi(\cdot)$ parameterized by weights and biases ϕ. These intermediate values are then used as inputs to the standard kernel resulting in the effective kernel $k_{DKL}(x, x') = k(g_\phi(x), g_\phi(x'))$. In order to learn the network weights and thereby learn representations of the data, it was proposed to maximize the marginal likelihood with respect to the weights ϕ along with the kernel hyperparameters. We denote all the hyperparameters by $\theta \coloneqq \{\phi, \sigma_n, \sigma_f, \{l_q\}_{q=1}^Q\}$.

3.3 STOCHASTIC VARIATIONAL DEEP KERNEL LEARNING

The straightforward deep kernel learning model suffers from two major drawbacks. First, due to the $\mathcal{O}(N^3)$ computational complexity of GPs, the standard DKL model suffers from poor scalability in the number of data.[1] Second, exact GP inference is only possible for Gaussian likelihoods, and therefore approximate techniques must be used for

[1] We note that Wilson et al. [2016a], which was the first paper to use the terminology "DKL", attempted to address scalability using KISS-GP [Wilson and Nickisch, 2015]; however, we use "DKL" to refer to the model with exact GP inference.



(a) Data fit (b) Complexity penalty

Figure 2: Training curves for the data fit and complexity penalties of the log marginal likelihood for the toy problem.

classification. To achieve both, we follow Bradshaw et al. [2017] in using stochastic variational inference (SVI) for GPs as introduced in Hensman et al. [2015], to result in stochastic variational DKL (SVDKL).[2]

Considering the case of C multiple outputs, we first introduce M latent inducing variables $\mathbf{u}_c = (u_{c1}, \ldots, u_{cM})^T$, indexed by M inducing inputs $z_m \in \mathbb{R}^Q$, which lie in the feature space at the output of the neural network. We assume the standard variational posterior over the inducing variables, $q(\mathbf{u}_c) = \mathcal{N}(\mathbf{m}_c, \mathbf{S}_c)$, leading to an approximate posterior $q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f}|\mathbf{u})q(\mathbf{u})$. We optimize the variational parameters $\mathbf{m}_c$ and $\mathbf{S}_c$, along with the model hyperparameters θ, jointly by maximizing the evidence lower bound (ELBO):

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{u})p(\mathbf{f}|\mathbf{u})}[\log p(\mathbf{y}|\mathbf{f})] - \mathrm{D}_{KL}(q(\mathbf{u})||p(\mathbf{u})). \quad (4)$$

Note that there are no restrictions on the likelihood $p(\mathbf{y}|\mathbf{f})$ as the first term can be estimated using Monte Carlo sampling with the reparameterization trick [Kingma and Welling, 2014, Rezende et al., 2014]. For Gaussian likelihoods and bounded inputs, theoretical results show that the ELBO can be made "tight" enough so it can be used as a stand-in for the marginal likelihood for hyperparameter optimization [Burt et al., 2020], if enough inducing points are given. Empirically, this has been shown to be the case for non-Gaussian likelihoods as well [Hensman et al., 2015].

4 PATHOLOGICAL BEHAVIOR IN A TOY PROBLEM

To motivate the rest of the paper, we first consider (exact) DKL on the toy problem from Snelson and Ghahramani [2006], a 1-dimensional regression problem consisting of 200 noisy input-output pairs generated from a GP with squared exponential kernel. We consider DKL using a two hidden-layer fully-connected ReLU network with layer widths $[100, 50]$ as the feature extractor, letting $Q = 2$ with a squared exponential kernel for the GP.[3] We describe the architecture and experimental details in more detail in Appendix B.

We plot the predictive posteriors of both a baseline GP with an SE kernel (corresponding to the ground truth), and DKL in Figures 1a and 1b, respectively. We observe that DKL suffers from pathological behavior: the fit is very jagged and extrapolates wildly outside the training data. On the other hand, the fit given by the SE kernel is smooth and fits the data well without any signs of overfitting. We therefore make the following observation:

Remark 1. *DKL models can be susceptible to overfitting, suggesting that the "complexity penalty" of the marginal likelihood may not always prevent overfitting.*

We next compare to the fit given by the deterministic neural network which uses the same feature extractor as the DKL model, so that both models have the same depth. To ensure a fair comparison, we retain the same training procedure, using the same learning rates, full batch training, and number of optimization steps, so that we only change the model and training loss (from the LML to mean squared error). We display the fit in Fig. 1c, which shows a nicer fit than the DKL fit of Fig. 1b: while there is some evidence of overfitting, it is in general much less than that of DKL. This leads us to our second observation:

Remark 2. *Surprisingly, DKL can exhibit* worse *overfitting than a standard neural network trained using maximum likelihood.*

Finally, we plot training curves from five different runs of DKL in Fig. 1d. From these, we observe that training is very unstable, with many significant spikes in the marginal likelihood objective. While we found that reducing the learning rate does improve stability, but only slightly (App. C.1). We also observe that each run often ends up settling in a different local minimum with very different final values of the log marginal likelihood. We plot different fits from different initializations in App. C.1, showing that these different local minima give very different fits with different generalization properties.

In general, this behavior is very concerning: one would hope that adding a Bayesian layer to a deterministic network would improve performance, as introducing Bayesian principles is often touted as a method to reduce overfitting (e.g. Osawa et al. [2019]). However, based off this toy problem performance seems to worsen with the addition of a Bayesian layer at the output. As this finding is seemingly at conflict with most of the literature, which has found that DKL, or variations thereof, can be useful, we devote the rest of this work to understanding when and why this pathological behavior arises, including for real datasets.

[2] We note again that this is slightly different in exact implementation to the SVDKL model proposed in Wilson et al. [2016b].

[3] We note that this is a smaller feature extractor than that proposed for a dataset of this size in Wilson et al. [2016a]

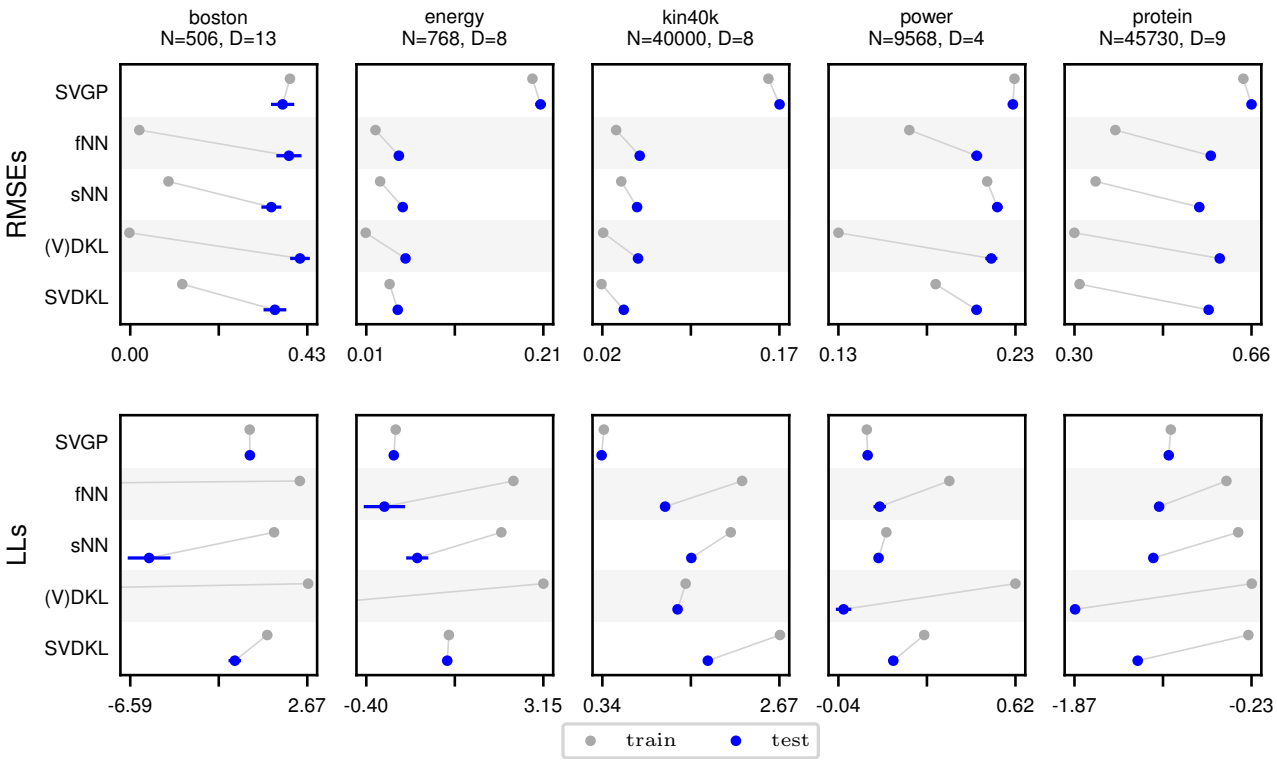


Figure 3: Results for the UCI datasets. We report train and test RMSEs and log likelihoods (LLs) for each method, averaged over the 20 splits. Left is better for RMSEs; right is better for LLs. Error bars represent one standard error.

5 UNDERSTANDING THE PATHOLOGY

To help understand the observed pathological behavior, we first look at the curves of the "data fit" and "complexity penalties" for five different initializations on the toy dataset. We present these curves in Fig. 2. We note that each of the data fit curves largely stabilize around -100 nats, so that the complexity terms seem to account for most of the differences in the final marginal likelihood (Fig. 1d). This behavior is explained by the following proposition, which states that the data fit term becomes uninteresting for any GPs with learnable signal variance trained on the marginal likelihood.

Proposition 1. *Consider the GP regression model as described in Eq. 1. Then, for any valid kernel function that can be written in the form $k(x, x') = \sigma_f^2 \hat{k}(x, x')$, where σ_f^2 is a learnable hyperparameter along with learnable noise σ_n^2 (and any other kernel hyperparameters), we have that the "data fit" term will equal $-N/2$ (where N is the number of datapoints) at the optimum of the marginal likelihood.*

The proof (App. A) is achieved by simple differentiation with respect to σ_f. This result is far-reaching, applying to the vast majority of kernel choices that we are aware of. This proposition therefore implies that the division of the marginal likelihood into "data fit" and "complexity penalty" terms is in general unhelpful, as the data fit term becomes uninteresting after training and the complexity penalty is responsible for any difference in marginal likelihood for GPs with different kernels.

However, we can still consider what a lower complexity penalty means for the learned kernel. Recall that the complexity penalty is given by

$$\frac{1}{2}\log|K + \sigma_n^2 I_N| = \frac{N}{2}\log\sigma_f^2 + \frac{1}{2}\log|\hat{K} + \hat{\sigma}_n^2 I_N|. \quad (5)$$

Maximizing the marginal likelihood encourages this term to be minimized, which can be done in at least two ways: minimizing σ_f, or minimizing the $|\hat{K} + \hat{\sigma}_n^2 I_N|$. However, there is little freedom in minimizing σ_f, because that would compromise the data fit. Therefore, the main mechanism for minimizing the complexity penalty would be through minimizing the second term. One way of doing this is to correlate the input points as much as possible: if there are enough degrees of freedom in the kernel, it is possible to "hack" the Gram matrix so that it can do this while minimizing the impact on the data fit term. We see this by looking at the correlation plots for the SE and DKL fits in Fig. 1: below the plots of the predictive posteriors, we have plotted correlation functions $\rho_{x'}(x) = k(x, x')/\sigma_f^2$ at two points x' given by the vertical dashed lines. We see that, while Fig. 1a shows the expected Gaussian bump for the SE kernel, Fig. 1b shows near-unity correlation functions for all values. Furthermore, in Appendix C.1 we show empirically that for fits that do not show as much correlation, the final

Table 1: LMLs/ELBOs per datapoint for UCI datasets.

	SVGP	(V)DKL	SVDKL
BOSTON	-1.66 $\pm$ 0.06	$\mathbf{2.47 \pm 0.00}$	0.47 $\pm$ 0.01
ENERGY	-0.07 $\pm$ 0.01	$\mathbf{3.01 \pm 0.02}$	1.21 $\pm$ 0.00
KIN40K	0.14 $\pm$ 0.00	1.41 $\pm$ 0.00	$\mathbf{2.62 \pm 0.00}$
POWER	0.01 $\pm$ 0.00	$\mathbf{0.57 \pm 0.00}$	0.25 $\pm$ 0.00
PROTEIN	-1.06 $\pm$ 0.00	$\mathbf{-0.32 \pm 0.01}$	-0.35 $\pm$ 0.00

marginal likelihood is worse, suggesting that increasing the correlation is indeed the main mechanism by which the model increases its marginal likelihood.

We summarize our findings in the remark:

Remark 3. *Adding flexibility to a GP can lead to pathological results, as the GP will use that flexibility to try to correlate* all *input points in the prior, not only the points where we would like correlations to appear.*

We now investigate how these observations relate to real, complex datasets, as well as to the prior literature which has shown that DKL can obtain good results.

6 DKL FOR REAL DATASETS

Despite these findings, multiple works have shown that DKL methods can perform well in practice [Wilson et al., 2016b, Bradshaw et al., 2017]. We now consider experiments on various datasets and architectures to further investigate the observed pathological behavior and how DKL succeeds. We provide full experimental details in Appendix B and additional experimental results in Appendix C.

6.1 DKL FOR UCI REGRESSION

We first consider DKL applied to a selection of regression datasets from the UCI repository [Dua and Graff, 2017]: BOSTON, ENERGY, KIN40K, POWER, PROTEIN. These represent a range of different sizes and dimensions: ENERGY, POWER, and PROTEIN were chosen specifically because we expect that they can benefit from the added depth to a GP [Salimbeni and Deisenroth, 2017].

We consider a range of different models, and we report train and test root mean square errors (RMSEs) and log likelihoods (LLs) in Fig. 3, and tabulate the log marginal likelihoods (LMLs) or ELBOs in Table 1. First, we consider a baseline stochastic variational GP (SVGP) model with an ARD SE kernel. As this is a GP model with few hyperparameters, we would not expect significant differences between training and testing performances. Indeed, looking at Fig. 3, this is exactly what we observe: the test performance is comparable to, and sometimes even slightly better than, the training performance for both RMSEs and LLs.

We compare to a neural network trained with mean squared error loss and DKL using the same neural network architecture for feature extractor (so that the depths are equal). We first consider DKL models where we use full-batch training, compared to a neural network with full-batch training, which we refer to as fNN. As full-batch training for DKL is expensive for larger datasets, for the KIN40K, POWER, and PROTEIN we instead use SVDKL trained with 1000 inducing points but full training batches, which we term *variational* DKL (VDKL). For both methods we use a small weight decay to help reduce overfitting, and we use the same number of gradient steps are used for each to ensure a fair comparison. Looking at the results for fNN and (V)DKL in Fig. 3, we see that both of these methods overfit quite drastically. This mirrors our observations in Remark 1 that DKL models can be susceptible to overfitting. In most cases the overfitting is noticeably worse for (V)DKL than it is for fNN, reflecting our observation in Remark 2. This is particularly concerning for the log likelihoods, as one would hope that the ability of DKL to express epistemic uncertainty through the last-layer GP would give it a major advantage over the neural network, which cannot do so.

In practice, however, many approaches for DKL and neural networks alike make use of stochastic minibatching during training. In fact, it is well-known that minibatch training induces implicit regularization for neural networks that helps generalization [Keskar et al., 2017]. We therefore investigate this for both DKL and neural networks: we refer to the stochastic minibatched network as sNN and compare to SVDKL, using the same batch sizes for both. Referring again to Fig. 3, we see that minibatching generally reduces overfitting compared to the full-batch versions, for both model types. Moreover, the difference between the full batch and stochastic minibatch performances of DKL seem to be greater than the corresponding differences for the standard neural networks, suggesting that the implicit regularization effect is stronger. The exception to this trend is KIN40K, which appears to be low-noise and simple for a deep model to predict for. We also note that with the exception of protein, SVDKL now performs the best of the deep models in terms of log likelihoods, and generally performs better than SVGP.

Finally, we consider Table 1, which shows the ELBOs/LMLs for each of the GP methods. SVGP has by far the worse ELBOs, whereas (V)DKL generally has by far the best. It is important to note that the ELBOs for SVDKL are worse than those for (V)DKL despite its generally better test performance. This suggests that improving the marginal likelihood for DKL models does not improve test performance, as one would desire for a Bayesian model. We summarize our findings in the following remark:

Remark 4. *The reason for DKL's successful performance is not an improved marginal likelihood, but rather that stochastic minibatching provides implicit regularization that protects against overfitting with the marginal likelihood.*

Table 2: Results for the UTKFace age regression task and CIFAR-10 classification, without data augmentation. We report means plus/minus one standard error, averaged over three runs.

	Batch size: 100					Batch size: 200/500		
	NN	SVDKL	pNN	fSVDKL	pSVDKL	pNN	fSVDKL	pSVDKL
UTKFace - ELBO	-	0.92 ± 0.01	-	1.05 ± 0.30	1.03 ± 0.10	-	0.75 ± 0.34	1.43 ± 0.04
Train RMSE	0.04 ± 0.00	0.04 ± 0.00	0.04 ± 0.00	0.08 ± 0.03	0.04 ± 0.00	0.04 ± 0.00	0.12 ± 0.03	0.04 ± 0.00
Test RMSE	0.40 ± 0.00	0.40 ± 0.01	0.41 ± 0.00	0.31 ± 0.07	0.38 ± 0.02	0.39 ± 0.01	0.23 ± 0.07	0.34 ± 0.02
Train LL	1.81 ± 0.01	1.30 ± 0.01	1.83 ± 0.01	1.16 ± 0.31	1.20 ± 0.08	1.83 ± 0.01	0.82 ± 0.34	1.60 ± 0.03
Test LL	-48.73 ± 1.64	-6.88 ± 0.38	-53.72 ± 1.71	-7.55 ± 3.42	-4.74 ± 1.35	-48.48 ± 2.07	-5.36 ± 4.78	-10.43 ± 2.94
CIFAR-10 - ELBO	-	-0.76 ± 0.28	-	-0.02 ± 0.00	-0.00 ± 0.00	-	-0.02 ± 0.00	-0.00 ± 0.00
Train Acc.	1.00 ± 0.00	0.76 ± 0.09	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Test Acc.	0.79 ± 0.00	0.63 ± 0.03	0.79 ± 0.00	0.78 ± 0.00	0.79 ± 0.00	0.79 ± 0.00	0.79 ± 0.00	0.79 ± 0.00
Train LL	-0.00 ± 0.00	-0.71 ± 0.28	-0.00 ± 0.00	-0.01 ± 0.00	-0.00 ± 0.00	-0.00 ± 0.00	-0.00 ± 0.00	-0.00 ± 0.00
Test LL	-2.05 ± 0.03	-1.37 ± 0.10	-2.30 ± 0.11	-1.14 ± 0.00	-1.13 ± 0.01	-2.88 ± 0.04	-1.07 ± 0.01	-1.45 ± 0.00
Inc. Test LL	-8.87 ± 0.10	-3.38 ± 0.77	-9.48 ± 0.30	-5.10 ± 0.01	-5.24 ± 0.05	-10.77 ± 0.07	-4.73 ± 0.03	-6.63 ± 0.03
ECE	0.18 ± 0.00	0.10 ± 0.05	0.19 ± 0.00	0.14 ± 0.00	0.15 ± 0.00	0.19 ± 0.00	0.13 ± 0.00	0.15 ± 0.00

Therefore, we observe again that the Bayesian benefits of the marginal likelihood do not apply in the overparameterized regime: indeed, we find that using the marginal likelihood can be worse than not being Bayesian at all.

6.2 DKL FOR IMAGE DATASETS

We now explore how these findings relate to high-dimensional, highly structure image datasets. We might expect that the benefits of DKL would be stronger for images than in the previous regression datasets, as the design of kernels for these high-dimensional spaces remains an open question despite numerous recent advances [van der Wilk et al., 2017, Dutordoir et al., 2020], and neural networks generally perform far better than kernel methods.

We first consider a regression problem using image inputs: an age regression task using the UTKFace dataset [Zhang et al., 2017]. The dataset consists of 23,708 images of size $200 \times 200 \times 3$ containing aligned and cropped faces. These images are annotated with age, gender and race, of which we focus on the task of predicting the subject's age.

We consider several models, all based on a ResNet-18 [He et al., 2016]: we take the standard ResNet-18 with 10-dimensional output, to which we add a ReLU nonlinearity and then either a linear output layer or an ARD SE GP, corresponding to the baseline neural network and SVDKL, respectively. We consider different feature widths Q in Appendix C. This construction ensures that both models have the same depth, so that any improvement observed for either cannot be attributed to the fact that the models have different depths. We consider the baseline neural network (NN) and SVDKL models. Additionally, as both Wilson et al. [2016b] and Bradshaw et al. [2017] use a pretraining and finetuning procedure for their models, we compare to this as well. We take the trained baseline NNs, and first learn the variational parameters and GP hyperparameters, keeping the network fixed. We refer to the result as the fixed net SVDKL (fSVDKL) model; we then train everything jointly for a number of epochs, resulting in the pretrained SVKDL (pSVDKL) model. Finally, so that any improvement for f/pSVDKL is not just from additional gradient steps, we also train the neural networks for the same number of epochs, resulting in the pretrained NN (pNN) model. We average all results over 3 independent runs using a batch size of 100, and we refer the reader to App. B.3 for full experimental details.

We report ELBOs, train and test RMSEs, and train and test log likelihoods for the normalized data in the top left portion of Table 2 (batch size 100). We see that SVDKL, the method without pretraining, obtains lower ELBOs than either fSVDKL or pSVDKL, which obtain largely similar ELBOs. We suspect that this is because of the difficulty in training large DKL models from scratch, as noted in Bradshaw et al. [2017]; this is also consistent with our earlier observation that training can be very unstable. We see that each method, except fSVDKL (with the fixed pretrained network), achieves similar train RMSE, but the test RMSEs are significantly worse, with fSVDKL obtaining the best. Unsurprisingly, the NN models perform poorly in terms of LL, as they are unable to express epistemic uncertainty. However, we also observe that additional training of the NNs worsens both test RMSEs and LLs. pSVDKL (where the network is allowed to change after pretraining) obtains the best test LL of all methods, as well as better test RMSE than the neural networks, showing that SVDKL can yield improvements consistent with the prior literature. We note, however, that there is still a substantial gap between train and test performance, indicating overfitting in a way consistent with Remark 1.

6.2.1 Increasing the Batch Size

From our UCI experiments, we hypothesized that implicit regularization from minibatch noise was key in obtaining good performance for SVDKL (Remark 4). We therefore consider increasing the batch size from 100 to 200 for the pretrained methods, keeping the pretrained neural networks the same; these results are also shown in the top right portion

Table 3: Results for the image datasets with data augmentation. We report means plus/minus one standard error, averaged over three runs.

	Batch size: 100				Batch size: 200/500		
	NN	pNN	fSVDKL	pSVDKL	pNN	fSVDKL	pSVDKL
UTKFace - ELBO	-	-	0.16 ± 0.03	0.14 ± 0.03	-	0.12 ± 0.06	0.45 ± 0.03
Train RMSE	0.19 ± 0.01	0.18 ± 0.00	0.19 ± 0.00	0.17 ± 0.01	0.13 ± 0.00	0.20 ± 0.01	0.12 ± 0.01
Test RMSE	0.36 ± 0.00	0.36 ± 0.00	0.36 ± 0.00	0.35 ± 0.00	0.35 ± 0.00	0.31 ± 0.04	0.35 ± 0.01
Train LL	0.25 ± 0.03	0.31 ± 0.01	0.25 ± 0.03	0.30 ± 0.03	0.65 ± 0.02	0.20 ± 0.06	0.63 ± 0.04
Test LL	-1.03 ± 0.07	-1.22 ± 0.05	-0.92 ± 0.07	-0.76 ± 0.03	-2.72 ± 0.21	-0.63 ± 0.30	-1.55 ± 0.17
CIFAR-10 - ELBO	-	-	-0.07 ± 0.00	-0.03 ± 0.00	-	-0.06 ± 0.01	-0.01 ± 0.00
Train Acc.	0.98 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	1.00 ± 0.00	0.98 ± 0.00	1.00 ± 0.00
Test Acc.	0.86 ± 0.00	0.86 ± 0.00	0.86 ± 0.00	0.86 ± 0.00	0.87 ± 0.00	0.86 ± 0.00	0.86 ± 0.00
Train LL	-0.05 ± 0.00	-0.02 ± 0.00	-0.05 ± 0.00	-0.03 ± 0.00	-0.01 ± 0.00	-0.05 ± 0.01	-0.01 ± 0.00
Test LL	-0.70 ± 0.01	-0.90 ± 0.00	-0.68 ± 0.00	-0.64 ± 0.00	-1.38 ± 0.03	-0.67 ± 0.02	-0.84 ± 0.00
Inc. Test LL	-4.83 ± 0.12	-6.31 ± 0.00	-4.65 ± 0.00	-4.58 ± 0.00	-8.97 ± 0.07	-4.66 ± 0.13	-6.06 ± 0.01
ECE	0.09 ± 0.00	0.11 ± 0.00	0.09 ± 0.00	0.09 ± 0.00	0.12 ± 0.00	0.09 ± 0.00	0.11 ± 0.00

of Table 2. We make a few key observations. First, this leads to a significantly improved ELBO for pSVDKL, which ends up helping the test RMSE. However, we see that instead of improving the test LL, it becomes significantly worse, whereas the train LL becomes better: clear evidence of overfitting. Moreover, fSVDKL, where the network is kept fixed, now outperforms pSVDKL, which has a better ELBO. Finally, we note that the behavior of pNN does not change significantly, in fact slightly improving with increased batch size: this suggests that the implicit regularization from minibatching is stronger for SVDKL than for standard NNs. All of these observations are consistent with our findings surrounding Remark 4, which argues that stochastic minibatching is crucial to the success of DKL methods, and a better marginal likelihood is associated with worse performance.

6.2.2 Image Classification

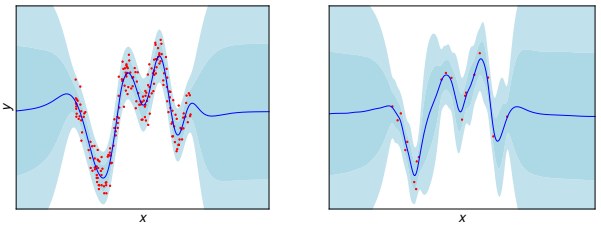
Our theory in Section 5 only applies directly to regression. As one of the main successes of current deep learning is in classification, it is therefore natural to wonder whether the trends we have observed also apply to classification tasks. We consider CIFAR-10 [Krizhevsky and Hinton, 2009], a popular dataset of $32 \times 32 \times 3$ images belonging to one of 10 classes. We again consider a modified ResNet-18 model, in which we have ensured that the depths remain the same between NN and DKL models. We consider training the models with batch sizes of 100 and 500. We look at ELBOs, accuracies, and LLs, as well as the LL for incorrectly classified test points, which can indicate overconfidence in predicting wrongly. We also look at expected calibration error (ECE; Guo et al. [2017]), a popular metric evaluating model calibration; results are shown in the lower portion of Table 2. Here, we see that plain SVDKL struggles even more to fit well, indicating the importance of pretraining. For the batch size 100 experiments, pSVDKL generally performs the best, reflecting the experience of Wilson et al. [2016b] and Bradshaw et al. [2017]. However, we again observe that increasing the batch size hurts pSVDKL, and fSVKDL outperforms it despite worse ELBOs.

6.3 DATA AUGMENTATION

It is common practice for image datasets to perform data augmentation, which effectively increases the size of the training dataset by using modified versions of the images. We briefly consider whether this affects our analysis by repeating the same experiments (without plain SVDKL, as it struggles to fit) with random cropping and horizontal flipping augmentations; see Table 3. Overall, we once again find that increasing the batch size still significantly hurts the performance of pSVDKL: whereas pSVDKL outperforms the fixed-network version for batch size 100, larger batch sizes reverse this, so that finetuning the network according to the ELBO hurts, rather than helps, performance. Therefore, in this case, using last-layer Bayesian inference is worse than not being Bayesian at all. These results reflect our findings in the previous remarks that using the marginal likelihood can be worse than using a standard likelihood, and that stochastic minibatching is one of the main reasons that DKL can be successful.

7 ADDRESSING THE PATHOLOGY

We have seen that the empirical Bayesian approach to overparameterized Gaussian processes can lead to pathological behavior. In particular, we have shown that methods that rely on the marginal likelihood to optimize a large number of hyperparameters can overfit, and that learning is unstable. While minibatching can help mitigate these issues, the overall performance is sensitive to the batch size, leading to a separate hyperparameter to tune. It is therefore natural to wonder whether we can address this by using a fully Bayesian approach, which has been shown to improve the predictive uncertainty of GP models [Lalchand and Rasmussen, 2020]. Indeed, Tran et al. [2019] showed that using Monte Carlo dropout to perform approximate Bayesian inference over the network parameters in DKL can improve calibration.



(a) Original dataset (b) Subsampled dataset

Figure 4: Predictive posteriors for fully Bayesian DKL using HMC for both the original toy dataset and the subsampled version from Titsias [2009].

Table 4: Results for the image datasets with SGLD.

	NN	SVDKL
UTKFace - Test RMSE	$\mathbf{0.16 \pm 0.00}$	$\mathbf{0.16 \pm 0.00}$
Test LL	0.39 ± 0.04	$\mathbf{0.42 \pm 0.03}$
CIFAR-10 - Test Acc.	$\mathbf{0.79 \pm 0.00}$	0.78 ± 0.00
Test LL	-1.89 ± 0.02	$\mathbf{-1.11 \pm 0.02}$
Inc. Test LL	-8.78 ± 0.11	$\mathbf{-4.94 \pm 0.10}$
ECE	0.18 ± 0.00	$\mathbf{0.13 \pm 0.00}$

We test this hypothesis using sampling-based methods. We first consider the toy problem from earlier, using HMC [Neal, 2011] to sample the neural network weights along with the other GP hyperparameters, using the marginal likelihood as the potential. We plot the resulting predictive posterior in Fig. 4a, and see that this completely resolves the problems observed earlier: in fact, the uncertainty in the outer regions is even greater than that given by the standard SE fit in Fig. 1a, while still concentrating where there is data. We additionally consider the subsampled version of the dataset, as introduced in Titsias [2009], in Fig. 4b. We see that there is still no overfitting despite the small dataset size: for a comparison to the baseline SE kernel and DKL, see Fig. 1 in the Appendix.

Unfortunately, HMC in its standard form does not scale to the larger datasets considered in Sec. 6.2, due to the necessity of calculating gradients over the entire dataset and the calculation of the acceptance probability. Therefore, we consider stochastic gradient Langevin dynamics (SGLD; Welling and Teh [2011]), which allows us to use mini-batches. We note that SGLD has relatively little additional training cost compared to SGD, as it simply injects scaled Gaussian noise into the gradients; the main cost is in memory and at test time. While we do not necessarily expect that this will be as accurate to the true posterior as HMC (see e.g. Johndrow et al. [2020]), we hope that it will give insights into what the performance of a fully Bayesian approach would be. We select a batch size of 100, and give test results for the NN and SVDKL for both UTKFace and CIFAR-10 without data augmentation in Table 4. We see that for both datasets, the additional uncertainty significantly helps the NN models. The improvement is significant for SVDKL for the UTKFace dataset, and while not so significant for CIFAR-10, we still observe slight improvements in log likelihoods and ECE, although at the expense of slightly lower test accuracy. Moreover, the fully Bayesian SVDKL outperforms the Bayesian NN in nearly every metric, and significantly so for the uncertainty-related metrics. In fact, for CIFAR-10, the original version of SVDKL (i.e. pSVDKL) outperforms the Bayesian NN for the uncertainty metrics, even for the larger batch size experiments. Therefore, we arrive at our final remark:

Remark 5. *A fully Bayesian approach to deep kernel learning can prevent overfitting and obtain the benefits of both neural networks and Gaussian processes.*

8 CONCLUSIONS

We have focused this work on exploring the performance of DKL in different regimes. We have shown that, while DKL models can achieve good performance, this is mostly because of implicit regularization due to stochastic minibatching rather than a better marginal likelihood. Based off our experiments, this stochastic regularization appears to be stronger than that for plain neural networks. Moreover, we have shown that when this stochastic regularization is limited, the performance can be worse than that of standard neural networks, with more overfitting and unstable training. This is surprising, because DKL models are more Bayesian than deterministic neural networks, and so one might expect that they would be less prone to overfitting due to the training objective being the marginal likelihood. However, we have shown that for highly parameterized models, the marginal likelihood is actually a poor objective, as it tries to correlate all the datapoints rather than those which should be correlated: therefore, a higher marginal likelihood does not improve performance as expected. This means that when the number of hyperparameters is large, the marginal likelihood cannot be relied upon for model selection, just as the standard maximum likelihood training loss cannot be used for model selection. Finally, we showed that a fully Bayesian approach to the neural network hyperparameters can overcome this limitation and improve the performance over the less Bayesian approach, fully showing the advantages of DKL models.

Acknowledgements

We would like to thank the anonymous reviewers, John Bradshaw, David R. Burt, Andrew Y.K. Foong, Markus Lange-Hegermann, James Requeima, Pola Schwöbel, and Andrew Gordon Wilson for helpful discussions and comments. SWO acknowledges the Gates Cambridge Trust for funding his doctoral studies.

References

John Bradshaw, Alexander G de G Matthews, and Zoubin Ghahramani. Adversarial examples, uncertainty, and transfer testing robustness in Gaussian process hybrid deep networks. *arXiv preprint arXiv:1707.02476*, 2017.

David R Burt, Carl Edward Rasmussen, and Mark van der Wilk. Convergence of sparse variational inference in gaussian processes regression. *Journal of Machine Learning Research*, 21(131):1–63, 2020.

Roberto Calandra, Jan Peters, Carl Edward Rasmussen, and Marc Peter Deisenroth. Manifold Gaussian processes for regression. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 3338–3345. IEEE, 2016.

Gavin C Cawley and Nicola LC Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *The Journal of Machine Learning Research*, 11:2079–2107, 2010.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Michael Dusenberry, Ghassen Jerfel, Yeming Wen, Yian Ma, Jasper Snoek, Katherine Heller, Balaji Lakshminarayanan, and Dustin Tran. Efficient and scalable Bayesian neural nets with rank-1 factors. In *International Conference on Machine Learning*, pages 2782–2792. PMLR, 2020.

Vincent Dutordoir, Mark van der Wilk, Artem Artemev, and James Hensman. Bayesian image classification with deep convolutional Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 1529–1539. PMLR, 2020.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE, 2016.

Jonathan Heek and Nal Kalchbrenner. Bayesian inference for large scale image classification. *arXiv preprint arXiv:1908.03491*, 2019.

James Hensman, Alexander G de G Matthews, and Zoubin Ghahramani. Scalable variational Gaussian process classification. In *International Conference on Artificial Intelligence and Statistics*, pages 351–360. PMLR, 2015.

James E Johndrow, Natesh S Pillai, and Aaron Smith. No free lunch for approximate MCMC. *arXiv preprint arXiv:2010.12514*, 2020.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Vidhi Lalchand and Carl Edward Rasmussen. Approximate inference for fully Bayesian Gaussian process regression. In *Symposium on Advances in Approximate Bayesian Inference*, pages 1–12. PMLR, 2020.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Jeremiah Z Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In *Advances in Neural Information Processing Systems*, 2020.

Radford M Neal. MCMC using hamiltonian dynamics. *Handbook of Markov chain Monte Carlo*, 2(11):2, 2011.

Sebastian W Ober and Laurence Aitchison. Global inducing point variational posteriors for Bayesian neural networks and deep Gaussian processes. *arXiv preprint arXiv:2005.08140*, 2020.

Sebastian W Ober and Carl Edward Rasmussen. Benchmarking the neural linear model for regression. *arXiv preprint arXiv:1912.08416*, 2019.

Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz Khan, Anirudh Jain, Runa Eschenhagen, Richard E Turner, and Rio Yokota. Practical deep learning with Bayesian principles. In *Advances in Neural Information Processing Systems*, pages 4289–4301, 2019.

Carl Edward Rasmussen and Christopher KI Williams. Gaussian Processes for Machine Learning. *ISBN-13 978-0-262-18253-9*, 2006.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286. PMLR, 2014.

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep Bayesian bandits showdown: An empirical comparison of Bayesian deep networks for Thompson sampling. In *International Conference on Learning Representations*, 2018.

Ruslan Salakhutdinov and Geoffrey Hinton. Using deep belief nets to learn covariance kernels for Gaussian processes. In *Advances in Neural Information Processing*, volume 7, pages 1249–1256, 2007.

Hugh Salimbeni and Marc Peter Deisenroth. Doubly stochastic variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 4588–4599, 2017.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, pages 1257–1264, 2006.

Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 567–574. PMLR, 2009.

Gia-Lac Tran, Edwin V Bonilla, John Cunningham, Pietro Michiardi, and Maurizio Filippone. Calibrating deep convolutional Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 1554–1563. PMLR, 2019.

Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning*, pages 9690–9700. PMLR, 2020.

Joost van Amersfoort, Lewis Smith, Andrew Jesson, Oscar Key, and Yarin Gal. Improving deterministic uncertainty estimation in deep learning for classification and regression. *arXiv preprint arXiv:2102.11409*, 2021.

Mark van der Wilk, Carl Edward Rasmussen, and James Hensman. Convolutional Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 2849–2858, 2017.

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning*, pages 681–688. PMLR, 2011.

Andrew Gordon Wilson and Hannes Nickisch. Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In *International Conference on Machine Learning*, pages 1775–1784. PMLR, 2015.

Andrew Gordon Wilson, Christoph Dann, and Hannes Nickisch. Thoughts on massively scalable Gaussian processes. *arXiv preprint arXiv:1511.01870*, 2015.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *International Conference on Artificial Intelligence and Statistics*, pages 370–378. PMLR, 2016a.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Stochastic variational deep kernel learning. In *Advances in Neural Information Processing Systems*, pages 2586–2594, 2016b.

Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient MCMC for Bayesian deep learning. In *International Conference on Learning Representations*, 2020.

Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017.