Bayesian Deep Learning and a Probabilistic Perspective of Generalization

Andrew Gordon Wilson **Pavel Izmailov**
New York University

Abstract

The key distinguishing property of a Bayesian approach is marginalization, rather than using a single setting of weights. Bayesian marginalization can particularly improve the accuracy and calibration of modern deep neural networks, which are typically underspecified by the data, and can represent many compelling but different solutions. We show that deep ensembles provide an effective mechanism for approximate Bayesian marginalization, and propose a related approach that further improves the predictive distribution by marginalizing within basins of attraction, without significant overhead. We also investigate the prior over functions implied by a vague distribution over neural network weights, explaining the generalization properties of such models from a probabilistic perspective. From this perspective, we explain results that have been presented as mysterious and distinct to neural network generalization, such as the ability to fit images with random labels, and show that these results can be reproduced with Gaussian processes. We also show that Bayesian model averaging alleviates double descent, resulting in monotonic performance improvements with increased flexibility. Finally, we provide a Bayesian perspective on tempering for calibrating predictive distributions.

1. Introduction

Imagine fitting the airline passenger data in Figure 1. Which model would you choose: (1) $f_1(x) = w_0 + w_1 x$, (2) $f_2(x) = \sum_{j=0}^{3} w_j x^j$, or (3) $f_3(x) = \sum_{j=0}^{10^4} w_j x^j$?

Put this way, most audiences overwhelmingly favour choices (1) and (2), for fear of overfitting. But of these options, choice (3) most honestly represents our beliefs. Indeed, it is likely that the ground truth explanation for the data is out of class for any of these choices, but there is some setting of the coefficients $\{w_j\}$ in choice (3) which provides a better description of reality than could be managed by choices (1) and (2), which are special cases of choice (3). Moreover, our beliefs about the generative processes for our observations, which are often very sophisticated, typically ought to be independent of how many data points we happen to observe.

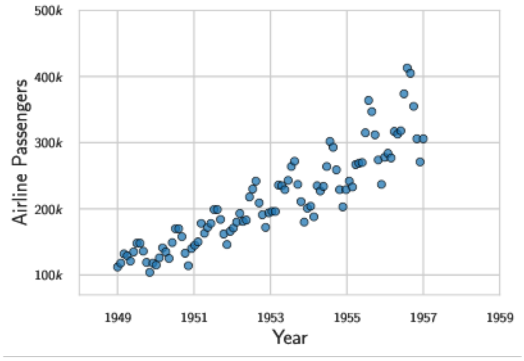


Figure 1. Airline passenger numbers recorded monthly.

And in modern practice, we are implicitly favouring choice (3): we often use neural networks with millions of parameters to fit datasets with thousands of points. Furthermore, non-parametric methods such as Gaussian processes often involve infinitely many parameters, enabling the flexibility for universal approximation (Rasmussen & Williams, 2006), yet in many cases provide very simple predictive distributions. Indeed, parameter counting is a poor proxy for understanding generalization behaviour.

From a probabilistic perspective, we argue that generalization depends largely on *two* properties, the *support* and the *inductive biases* of a model. Consider Figure 2(a), where on the horizontal axis we have a conceptualization of all possible datasets, and on the vertical axis the Bayesian *evidence* for a model. The evidence, or marginal likelihood, $p(\mathcal{D}|\mathcal{M}) = \int p(\mathcal{D}|\mathcal{M}, w)p(w)dw$, is the probability we would generate a dataset if we were to randomly sample from the prior over functions $p(f(x))$ induced by a prior over parameters $p(w)$. We define the support as the range of datasets for which $p(\mathcal{D}|\mathcal{M}) > 0$. We define the inductive biases as the relative prior probabilities of different datasets — the *distribution of support* given by $p(\mathcal{D}|\mathcal{M})$. A similar schematic to Figure 2(a) was used by MacKay (1992) to understand an Occam's razor effect in using the evidence for model selection; we believe it can also be used to reason about model construction and generalization.

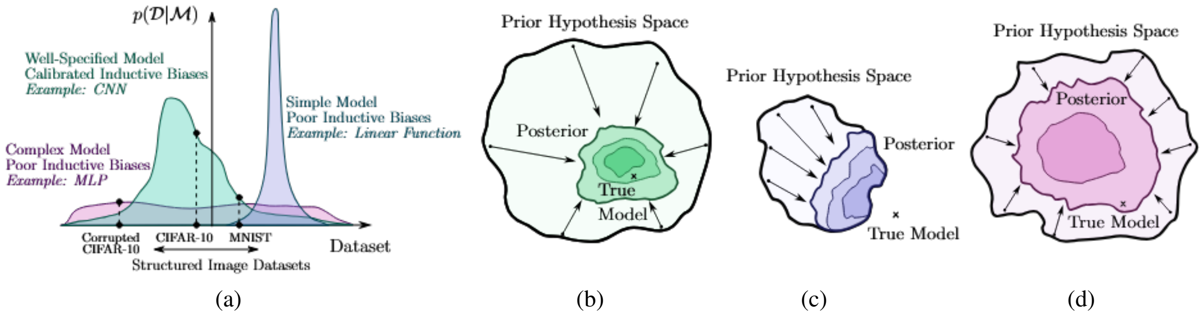


Figure 2. **A probabilistic perspective of generalization.** (a) Ideally, a model supports a wide range of datasets, but with inductive biases that provide high prior probability to a particular class of problems being considered. Here, the CNN is preferred over the linear model and the fully-connected MLP for CIFAR-10 (while we do not consider MLP models to in general have poor inductive biases, here we are considering a hypothetical example involving images and a very large MLP). (b) By representing a large hypothesis space, a model can contract around a true solution, which in the real-world is often very sophisticated. (c) With truncated support, a model will converge to an erroneous solution. (d) Even if the hypothesis space contains the truth, a model will not efficiently contract unless it also has reasonable inductive biases.

From this perspective, we want the support of the model to be large so that we can represent any hypothesis we believe to be possible, even if it is unlikely. We would even want the model to be able to represent pure noise, such as noisy CIFAR (Zhang et al., 2016), as long as we honestly believe there is some non-zero, but potentially arbitrarily small, probability that the data are simply noise. Crucially, we also need the inductive biases to carefully represent which hypotheses we believe to be a priori likely for a particular problem class. If we are modelling images, then our model should have statistical properties, such as convolutional structure, which are good descriptions of images.

Figure 2(a) illustrates three models. We can imagine the blue curve as a simple linear function, $f(x) = w_0 + w_1 x$, combined with a distribution over parameters $p(w_0, w_1)$, e.g., $\mathcal{N}(0, I)$, which induces a distribution over functions $p(f(x))$. Parameters we sample from our prior $p(w_0, w_1)$ give rise to functions $f(x)$ that correspond to straight lines with different slopes and intercepts. This model thus has truncated support: it cannot even represent a quadratic function. But because the marginal likelihood must normalize over datasets $\mathcal{D}$, this model assigns much mass to the datasets it does support. The red curve could represent a large fully-connected MLP. This model is highly flexible, but distributes its support across datasets too evenly to be particularly compelling for many image datasets. The green curve could represent a convolutional neural network, which represents a compelling specification of support and inductive biases for image recognition: this model has the flexibility to represent many solutions, but its structural properties provide particularly good support for many image problems.

With large support, we cast a wide enough net that the posterior can contract around the true solution to a given problem as in Figure 2(b), which in reality we often believe to be very sophisticated. On the other hand, the simple model will have a posterior that contracts around an erroneous solution if it is not contained in the hypothesis space as in Figure 2(c). Moreover, in Figure 2(d), the model has wide support, but does not contract around a good solution because its support is too evenly distributed.

Returning to the opening example, we can justify the high order polynomial by wanting large support. But we would still have to carefully choose the prior on the coefficients to induce a distribution over functions that would have reasonable inductive biases. Indeed, this Bayesian notion of generalization is not based on a single number, but is a two dimensional concept. From this probabilistic perspective, it is crucial not to conflate the *flexibility* of a model with the *complexity* of a model class. Indeed Gaussian processes with RBF kernels have large support, and are thus flexible, but have inductive biases towards very simple solutions. We also see that *parameter counting* has no significance in this perspective of generalization: what matters is how a distribution over parameters combines with a functional form of a model, to induce a distribution over solutions. Rademacher complexity (Mohri & Rostamizadeh, 2009), VC dimension (Vapnik, 1998), and many conventional metrics, are by contrast *one dimensional notions*, corresponding roughly to the support of the model, which is why they have been found to provide an incomplete picture of generalization in deep learning (Zhang et al., 2016).

In this paper we reason about Bayesian deep learning from a probabilistic perspective of generalization. The key distinguishing property of a Bayesian approach is marginalization instead of optimization, where we represent solutions given by all settings of parameters weighted by their posterior probabilities, rather than bet everything on a single setting of parameters. Neural networks are typically underspecified by the data, and can represent many different but high performing models corresponding to different settings of parameters, which is exactly when marginalization will make the biggest difference for accuracy and calibration. Moreover, we clarify that the recent deep ensembles (Lakshminarayanan et al., 2017) are not a competing approach to Bayesian inference, but can be viewed as a compelling mechanism for Bayesian marginalization. Indeed, we empirically demonstrate that deep ensembles can provide a *better* approximation to the Bayesian predictive distribution than standard Bayesian approaches. We further propose a new method, MultiSWAG, inspired by deep ensembles, which marginalizes within basins of attraction — achieving significantly improved performance, with a similar training time.

We then investigate the properties of priors over functions induced by priors over the weights of neural networks, showing that they have reasonable inductive biases. We also show that the mysterious generalization properties recently presented in Zhang et al. (2016) can be understood by reasoning about prior distributions over functions, and are not specific to neural networks. Indeed, we show Gaussian processes can also perfectly fit images with random labels, yet generalize on the noise-free problem. These results are a consequence of large support but reasonable inductive biases for common problem settings. We further show that while Bayesian neural networks can fit the noisy datasets, the marginal likelihood has much better support for the noise free datasets, in line with Figure 2. We additionally show that the multimodal marginalization in MultiSWAG alleviates double descent, so as to achieve monotonic improvements in performance with model flexibility, in line with our perspective of generalization. MultiSWAG also provides significant improvements in both accuracy and NLL over SGD training and unimodal marginalization. Finally we provide several perspectives on tempering in Bayesian deep learning.

In the Appendix we provide several additional experiments and results. We also provide code at `https://github.com/izmailovpavel/understandingbdl`.

2. Related Work

Notable early works on Bayesian neural networks include MacKay (1992), MacKay (1995), and Neal (1996). These works generally argue in favour of making the model class for Bayesian approaches as flexible as possible, in line with Box & Tiao (1973). Accordingly, Neal (1996) pursued the limits of large Bayesian neural networks, showing that as the number of hidden units approached infinity, these models become Gaussian processes with particular kernel functions. This work harmonizes with recent work describing the neural tangent kernel (e.g., Jacot et al., 2018).

The marginal likelihood is often used for Bayesian hypothesis testing, model comparison, and hyperparameter tuning, with *Bayes factors* used to select between models (Kass & Raftery, 1995). MacKay (2003, Ch. 28) uses a diagram similar to Fig 2(a) to show the marginal likelihood has an *Occam's razor* property, favouring the simplest model consistent with a given dataset, even if the prior assigns equal probability to the various models. Rasmussen & Ghahramani (2001) reasons about how the marginal likelihood can favour large flexible models, as long as such models correspond to a reasonable distribution over functions.

There has been much recent interest in developing Bayesian approaches for modern deep learning, with new challenges and architectures quite different from what had been considered in early work. Recent work has largely focused on scalable inference (e.g., Blundell et al., 2015; Gal & Ghahramani, 2016; Kendall & Gal, 2017; Ritter et al., 2018; Khan et al., 2018; Maddox et al., 2019), function-space inspired priors (e.g., Yang et al., 2019; Louizos et al., 2019; Sun et al., 2019; Hafner et al., 2018), and developing flat objective priors in parameter space, directly leveraging the biases of the neural network functional form (e.g, Nalisnick, 2018). Wilson (2020) provides a note motivating Bayesian deep learning.

Early works tend to provide a connection between loss geometry and generalization using minimum description length frameworks (e.g., Hinton & Van Camp, 1993; Hochreiter & Schmidhuber, 1997; MacKay, 1995). Empirically, Keskar et al. (2016) argue that smaller batch SGD provides better generalization than large batch SGD, by finding flatter minima. Chaudhari et al. (2019) and Izmailov et al. (2018) design optimization procedures to specifically find flat minima.

By connecting flat solutions with ensemble approximations, Izmailov et al. (2018) also suggest that functions associated with parameters in flat regions ought to provide different predictions on test data, for flatness to be helpful in generalization, which is distinct from the flatness in Dinh et al. (2017). Garipov et al. (2018) also show that there are mode connecting curves, forming loss valleys, which contain a variety of distinct solutions. We argue that flat regions of the loss containing a diversity of solutions is particularly relevant for Bayesian model averaging, since the model average will then contain many compelling and complementary explanations for the data. Additionally, Huang et al. (2019) describes neural networks as having a *blessing of dimensionality*, since flat regions will occupy much greater volume in a high dimensional space, which we argue means that flat solutions will dominate in the Bayesian model average.

Smith & Le (2018) and MacKay (2003, Chapter 28) additionally connect the width of the posterior with Occam factors; from a Bayesian perspective, larger width corresponds to a smaller Occam factor, and thus ought to provide better generalization. Dziugaite & Roy (2017) and Smith & Le (2018) also provide different perspectives on the results in Zhang et al. (2016), which shows that deep convolutional neural networks can fit CIFAR-10 with random labels and no training error. The PAC-Bayes bound of Dziugaite & Roy (2017) becomes vacuous when applied to randomly-labelled binary MNIST. Smith & Le (2018) show that logistic regression can fit noisy labels on sub-sampled MNIST, interpreting the result from an Occam factor perspective.

In general, PAC-Bayes provides a compelling framework for deriving explicit non-asymptotic generalization bounds for stochastic networks with distributions over parameters (McAllester, 1999; Langford & Caruana, 2002; Dziugaite & Roy, 2017; Neyshabur et al., 2017; 2018; Masegosa, 2019; Jiang et al., 2019; Guedj, 2019; Alquier, 2021). Langford & Caruana (2002) devised a PAC-Bayes generalization bound for small stochastic neural networks (two layer with two hidden units) achieving an improvement over the existing deterministic generalization bounds. Dziugaite & Roy (2017) extended this approach, optimizing a PAC-Bayes bound with respect to a parametric distribution over the weights of the network, exploiting the flatness of solutions discovered by SGD, for non-vacuous bounds with an overparametrized network on binary MNIST. Neyshabur et al. (2017) also discuss the connection between PAC-Bayes bounds and sharpness, and Neyshabur et al. (2018) devises PAC-Bayes bounds based on spectral norms of the layers and the Frobenius norm of the weights of the network. Achille & Soatto (2018) additionally combine PAC-Bayes and information theoretic approaches to argue that flat minima have low information content. Masegosa (2019) also proposes variational and ensemble learning methods based on PAC-Bayes analysis under model misspecification. Jiang et al. (2019) provide a review and comparison of several generalization bounds, including PAC-Bayes.

Our contributions are largely orthogonal and complementary to PAC-Bayes. PAC-Bayes bounds can be be improved by, e.g. fewer parameters, and very compact priors, which can be different from what provides optimal generalization. From our perspective, model flexibility and priors with *large* support, rather than compactness, are desirable. Moreover, we show the great significance of multi-basin marginalization for generalization, whereas multi-basin marginalization has a minimal logarithmic effect on PAC-Bayes bounds. Indeed, *marginalization* — a posterior weighted model average — is our key focus, whereas PAC-Bayes bounds are typically bounding the empirical risk of a single sample. In general, our focus is complementary to PAC-Bayes, aiming to provide *prescriptive* intuitions on model construction, inference, and neural network priors, as well as new connections between Bayesian model averaging and deep ensembles, benefits of Bayesian model averaging in the context of modern deep neural networks, views of marginalization that contrast with simple Monte Carlo, and new methods for Bayesian marginalization in deep learning.

In other work, Pearce et al. (2018) propose a modification of deep ensembles and argue that it performs approximate Bayesian inference, and Gustafsson et al. (2019) briefly mention how deep ensembles can be viewed as samples from an approximate posterior. In the context of deep ensembles, we believe it is natural to consider the BMA integral separately from the simple Monte Carlo approximation that is often used to approximate this integral; to compute an accurate predictive distribution, we do not need samples from a posterior, or even a faithful approximation to the posterior.

Fort et al. (2019) considered the diversity of predictions produced by models from multiple independent SGD runs, and suggested to ensemble averages of SGD iterates. Although MultiSWA (one of the methods considered in Section 4) is related to this idea, the crucial practical difference is that MultiSWA uses a learning rate schedule that selects for flat regions of the loss, the key to the success of the SWA method (Izmailov et al., 2018). Section 4 also shows that MultiSWAG, which we propose for multimodal Bayesian marginalization, outperforms MultiSWA.

Double descent, which describes generalization error that decreases, increases, and then again decreases with model flexibility, was demonstrated early by Opper et al. (1990). Recently, Belkin et al. (2019) extensively demonstrated double descent, leading to a surge of modern interest, with Nakkiran et al. (2019) showing double descent in deep learning. Nakkiran et al. (2020) shows that tuned l_2 regularization can mitigate double descent. Alternatively, we show that Bayesian model averaging, particularly based on multimodal marginalization, can alleviate even prominent double descent behaviour.

Tempering in Bayesian modelling has been considered under the names *Safe Bayes*, *generalized Bayesian inference*, and *fractional Bayesian inference* (e.g., de Heide et al., 2019; Grünwald et al., 2017; Barron & Cover, 1991; Walker & Hjort, 2001; Zhang, 2006; Bissiri et al., 2016; Grünwald, 2012). We provide several perspectives of tempering in Bayesian deep learning, and analyze the results in a recent paper by Wenzel et al. (2020) that questions tempering for Bayesian neural networks.

3. Bayesian Marginalization

Often the predictive distribution we want to compute is given by

$$p(y|x, \mathcal{D}) = \int p(y|x, w)p(w|\mathcal{D})dw\,. \qquad (1)$$

The outputs are y (e.g., regression values, class labels, ...), indexed by inputs x (e.g. spatial locations, images, ...), the weights (or parameters) of the neural network $f(x; w)$ are w, and $\mathcal{D}$ are the data. Eq. (1) represents a *Bayesian model average* (BMA). Rather than bet everything on one hypothesis — with a single setting of parameters w — we want to use all settings of parameters, weighted by their posterior probabilities. This procedure is called *marginalization* of the parameters w, as the predictive distribution of interest no longer conditions on w. This is not a controversial equation, but simply the sum and product rules of probability.

3.1. Importance of Marginalization in Deep Learning

In general, we can view classical training as performing approximate Bayesian inference, using the approximate posterior $p(w|\mathcal{D}) \approx \delta(w = \hat{w})$ to compute Eq. (1), where δ is a Dirac delta function that is zero everywhere except at $\hat{w} = \text{argmax}_w p(w|\mathcal{D})$. In this case, we recover the standard predictive distribution $p(y|x, \hat{w})$. From this perspective, many alternatives, albeit imperfect, will be preferable — including impoverished Gaussian posterior approximations for $p(w|\mathcal{D})$, even if the posterior or likelihood are actually highly non-Gaussian and multimodal.

The difference between a classical and Bayesian approach will depend on how sharp the posterior $p(w|\mathcal{D})$ becomes. If the posterior is sharply peaked, and the conditional predictive distribution $p(y|x, w)$ does not vary significantly where the posterior has mass, there may be almost no difference, since a delta function may then be a reasonable approximation of the posterior for the purpose of BMA. However, modern neural networks are usually highly underspecified by the available data, and therefore have diffuse likelihoods $p(\mathcal{D}|w)$, not strongly favouring any one setting of parameters. Not only are the likelihoods diffuse, but different settings of the parameters correspond to a diverse variety of compelling hypotheses for the data (Garipov et al., 2018; Izmailov et al., 2019). This is exactly the setting when we *most* want to perform a Bayesian model average, which will lead to an ensemble containing many different but high performing models, for better calibration *and* accuracy than classical training.

Loss Valleys. Flat regions of low loss (negative log posterior density $-\log p(w|\mathcal{D})$) are associated with good generalization (e.g., Hochreiter & Schmidhuber, 1997; Hinton & Van Camp, 1993; Dziugaite & Roy, 2017; Izmailov et al., 2018; Keskar et al., 2016). While flat solutions that generalize poorly can be contrived through reparametrization (Dinh et al., 2017), the flat regions that lead to good generalization contain a *diversity* of high performing models on test data (Izmailov et al., 2018), corresponding to different parameter settings in those regions. And indeed, there are large contiguous regions of low loss that contain such solutions, even connecting together different SGD solutions (Garipov et al., 2018; Izmailov et al., 2019) (see also Figure 11, Appendix).

Since these regions of the loss represent a large volume in a high-dimensional space (Huang et al., 2019), and provide a diversity of solutions, they will dominate in forming the predictive distribution in a Bayesian model average. By contrast, if the parameters in these regions provided similar functions, as would be the case in flatness obtained through reparametrization, these functions would be redundant in the model average. That is, although the solutions of high posterior density can provide poor generalization, it is the solutions that generalize well that will have greatest posterior *mass*, and thus be automatically favoured by the BMA.

Calibration by Epistemic Uncertainty Representation. It has been noticed that modern neural networks are often *miscalibrated* in the sense that their predictions are typically *overconfident* (Guo et al., 2017). For example, in classification the highest softmax output of a convolutional neural network is typically much larger than the probability of the associated class label. The fundamental reason for miscalibration is ignoring epistemic uncertainty. A neural network can represent many models that are consistent with our observations. By selecting only one, in a classical procedure, we lose uncertainty when the models disagree for a test point. In regression, we can visualize epistemic uncertainty by looking at the spread of the predictive distribution; as we move away from the data, there are a greater variety of consistent solutions, leading to larger uncertainty, as in Figure 4. We can further calibrate the model with tempering, which we discuss in the Appendix Section 8.

Accuracy. An often overlooked benefit of Bayesian model averaging in *modern* deep learning is improved *accuracy*. If we average the predictions of many high performing models that disagree in some cases, we should see significantly improved accuracy. This benefit is now starting to be observed in practice (e.g., Izmailov et al., 2019). Improvements in accuracy are very convincingly exemplified by *deep ensembles* (Lakshminarayanan et al., 2017), which have been perceived as a competing approach to Bayesian methods, but in fact provides a compelling mechanism for approximate Bayesian model averaging, as we show in Section 3.3. We also demonstrate significant accuracy benefits for multimodal Bayesian marginalization in Section 7.

3.2. Beyond Monte Carlo

Nearly all approaches to estimating the integral in Eq. (1), when it cannot be computed in closed form, involve a *simple Monte Carlo* approximation: $p(y|x, \mathcal{D}) \approx \frac{1}{J}\sum_{j=1}^{J} p(y|x, w_j)\,, w_j \sim p(w|\mathcal{D})$. In practice, the samples from the posterior $p(w|\mathcal{D})$ are also approximate, and found through MCMC or deterministic methods. The deterministic methods approximate $p(w|\mathcal{D})$ with a different more convenient density $q(w|\mathcal{D}, \theta)$ from which we can sample, often chosen to be Gaussian. The parameters θ are selected to make q close to p in some sense; for example, variational approximations (e.g., Beal, 2003), which have emerged as a popular deterministic approach, find $\text{argmin}_{\theta}\mathcal{KL}(q||p)$. Other standard deterministic approximations include Laplace (e.g., MacKay, 1995), EP (Minka, 2001a), and INLA (Rue et al., 2009).

From the perspective of estimating the predictive distribution in Eq. (1), we can view simple Monte Carlo as approximating the posterior with a set of point masses, with locations given by samples from another approximate posterior q, even if q is a continuous distribution. That is, $p(w|\mathcal{D}) \approx \sum_{j=1}^{J} \delta(w = w_j)\,, w_j \sim q(w|\mathcal{D})$.

Ultimately, the goal is to accurately compute the predictive distribution in Eq. (1), rather than find a generally accurate representation of the posterior. In particular, we must carefully represent the posterior in regions that will make the greatest contributions to the BMA integral. In terms of efficiently computing the predictive distribution, we do not necessarily want to place point masses at locations given by samples from the posterior. For example, functional diversity is important for a good approximation to the BMA integral, because we are summing together terms of the form $p(y|x, w)$; if two settings of the weights w_i and w_j each provide high likelihood (and consequently high posterior density), but give rise to similar functions $f(x; w_i)$, $f(x; w_j)$, then they will be largely redundant in the model average, and the second setting of parameters will not contribute much to estimating the BMA integral for the unconditional predictive distribution. In Sections 3.3 and 4, we consider how various approaches approximate the predictive distribution.

3.3. Deep Ensembles are BMA

Deep ensembles (Lakshminarayanan et al., 2017) is fast becoming a gold standard for accurate and well-calibrated predictive distributions. Recent reports (e.g., Ovadia et al., 2019; Ashukha et al., 2020) show that deep ensembles appear to outperform some particular approaches to Bayesian neural networks for uncertainty representation, leading to the confusion that deep ensembles and Bayesian methods are competing approaches. These methods are often explicitly referred to as non-Bayesian (e.g., Lakshminarayanan et al., 2017; Ovadia et al., 2019; Wenzel et al., 2020). To the contrary, we argue that deep ensembles are actually a compelling approach to Bayesian model averaging, in the vein of Section 3.2.

There is a fundamental difference between a Bayesian model average and some approaches to ensembling. The Bayesian model average assumes that *one* hypothesis (one parameter setting) is correct, and averages over models due to an inability to distinguish between hypotheses given limited information (Minka, 2000). As we observe more data, the posterior collapses onto a single hypothesis. If the true explanation for the data is a combination of hypotheses, then the Bayesian model average may appear to perform worse as we observe more data. Some ensembling methods work by enriching the hypothesis space, and therefore do not collapse in this way. Deep ensembles, however, are formed by MAP or maximum likelihood retraining of the same architecture multiple times, leading to different basins of attraction. The deep ensemble will therefore collapse in the same way as a Bayesian model average, as the posterior concentrates. Since the hypotheses space (support) for a modern neural network is large, containing many different possible explanations for the data, posterior collapse will often be desirable.

Furthermore, by representing multiple basins of attraction, deep ensembles can provide a *better* approximation to the BMA than the Bayesian approaches in Ovadia et al. (2019). Indeed, the functional diversity is important for a good approximation to the BMA integral, as per Section 3.2. The approaches referred to as Bayesian in Ovadia et al. (2019) instead focus their approximation on a single basin, which may contain a lot of redundancy in function space, making a relatively minimal contribution to computing the Bayesian predictive distribution. On the other hand, retraining a neural network multiple times for deep ensembles incurs a significant computational expense. The single basin approaches may be preferred if we are to control for computation. We explore these questions in Section 4.

4. An Empirical Study of Marginalization

We have shown that deep ensembles can be interpreted as an approximate approach to Bayesian marginalization, which selects for functional diversity by representing multiple basins of attraction in the posterior. Most Bayesian deep learning methods instead focus on faithfully approximating a posterior within a single basin of attraction. We propose a new method, MultiSWAG, which combines these two types of approaches. MultiSWAG combines multiple independently trained SWAG approximations (Maddox et al., 2019), to create a mixture of Gaussians approximation to the posterior, with each Gaussian centred on a different basin of attraction. We note that MultiSWAG does not require *any*

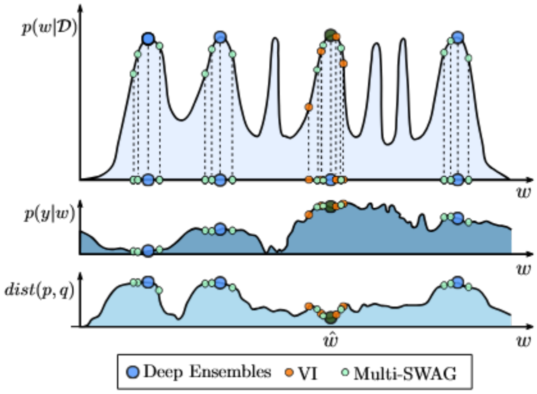


Figure 3. **Approximating the BMA.** $p(y|x,\mathcal{D}) = \int p(y|x,w)p(w|\mathcal{D})dw$. **Top:** $p(w|\mathcal{D})$, with representations from VI (orange) deep ensembles (blue), MultiSWAG (red). **Middle:** $p(y|x,w)$ as a function of w for a test input x. This function does not vary much within modes, but changes significantly between modes. **Bottom:** Distance between the true predictive distribution and the approximation, as a function of representing a posterior at an additional point w, assuming we have sampled the mode in dark green. There is more to be gained by exploring new basins, than continuing to explore the same basin.

additional training time over standard deep ensembles.

We illustrate the conceptual difference between deep ensembles, a standard variational single basin approach, and MultiSWAG, in Figure 3. In the top panel, we have a conceptualization of a multimodal posterior. VI approximates the posterior with multiple samples within a single basin. But we see in the middle panel that the conditional predictive distribution $p(y|x,w)$ does not vary significantly within the basin, and thus each additional sample contributes minimally to computing the marginal predictive distribution $p(y|x,\mathcal{D})$. On the other hand, $p(y|x,w)$ varies significantly between basins, and thus each point mass for deep ensembles contributes significantly to the marginal predictive distribution. By sampling within the basins, MultiSWAG provides additional contributions to the predictive distribution. In the bottom panel, we have the gain in approximating the predictive distribution when adding a point mass to the representation of the posterior, as a function of its location, assuming we have already sampled the mode in dark green. Including samples from different modes provides significant gain over continuing to sample from the same mode, and including weights in wide basins provide relatively more gain than the narrow ones.

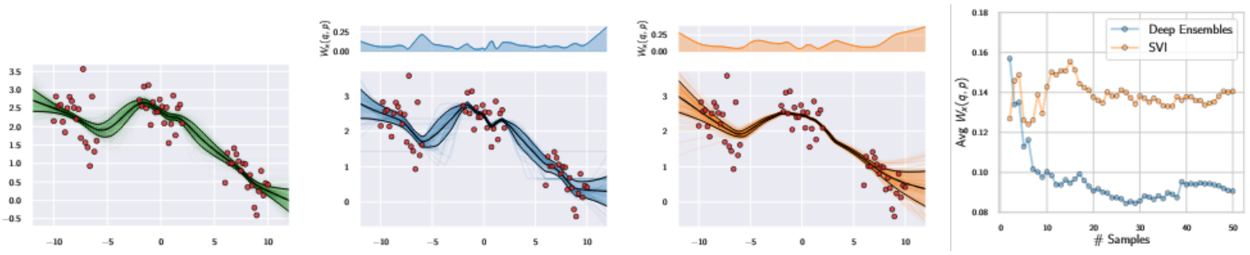
In Figure 4 we evaluate single basin and multi-basin approaches in a case where we can near-exactly compute the predictive distribution. We provide details for generating the data and training the models in Appendix D.1. We see that the predictive distribution given by deep ensembles is qualitatively closer to the true distribution, compared to the single basin variational method: between data clusters, the deep ensemble approach provides a similar representation of epistemic uncertainty, whereas the variational method is extremely overconfident in these regions. Moreover, we see that the Wasserstein distance between the true predictive distribution and these two approximations quickly shrinks with number of samples for deep ensembles, but is roughly independent of number of samples for the variational approach. Thus the deep ensemble is providing a better approximation of the Bayesian model average in Eq. (1) than the single basin variational approach, which has traditionally been labelled as the Bayesian alternative.

Next, we evaluate MultiSWAG under distribution shift on the CIFAR-10 dataset (Krizhevsky et al., 2014), replicating the setup in Ovadia et al. (2019). We consider 16 data corruptions, each at 5 different levels of severity, introduced by Hendrycks & Dietterich (2019). For each corruption, we evaluate the performance of deep ensembles and MultiSWAG varying the training budget. For deep ensembles we show performance as a function of the number of independently trained models in the ensemble. For MultiSWAG we show performance as a function of the number of independent SWAG approximations that we construct; we then sample 20 models from each of these approximations to construct the final ensemble.

While the training time for MultiSWAG is the same as for deep ensembles, at test time MultiSWAG is more expensive, as the corresponding ensemble consists of a larger number of models. To account for situations when test time is constrained, we also propose MultiSWA, a method that ensembles independently trained SWA solutions (Izmailov et al., 2018). SWA solutions are the means of the corresponding Gaussian SWAG approximations. Izmailov et al. (2018) argue that SWA solutions approximate the local ensembles represented by SWAG with a single model.

In Figure 5 we show the negative log-likelihood as a function of the number of independently trained models for a Preactivation ResNet-20 on CIFAR-10 corrupted with Gaussian blur with varying levels of intensity (increasing from left to right) in Figure 5. MultiSWAG outperforms deep ensembles significantly on highly corrupted data. For lower levels of corruption, MultiSWAG works particularly well when only a small number of independently trained models are available. We note that MultiSWA also outperforms deep ensembles, and has the same computational requirements at training and test time as deep ensembles. We present results for other types of corruption in Appendix Figures 14, 15, 16, 17, showing similar trends. In general, there is an extensive evaluation of MultiSWAG in the Appendix.

Our perspective of generalization is deeply connected with Bayesian marginalization. In order to best realize the bene-



(a) Exact (b) Deep Ensembles (c) Variational Inference (d) Discrepancy with True BMA

Figure 4. **Approximating the true predictive distribution. (a)**: A close approximation of the true predictive distribution obtained by combining 200 HMC chains. **(b)**: Deep ensembles predictive distribution using 50 independently trained networks. **(c)**: Predictive distribution for factorized variational inference (VI). **(d)**: Convergence of the predictive distributions for deep ensembles and variational inference as a function of the number of samples; we measure the average Wasserstein distance between the marginals in the range of input positions. The multi-basin deep ensembles approach provides a more faithful approximation of the Bayesian predictive distribution than the conventional single-basin VI approach, which is overconfident between data clusters. The top panels show the Wasserstein distance between the true predictive distribution and the deep ensemble and VI approximations, as a function of inputs x.

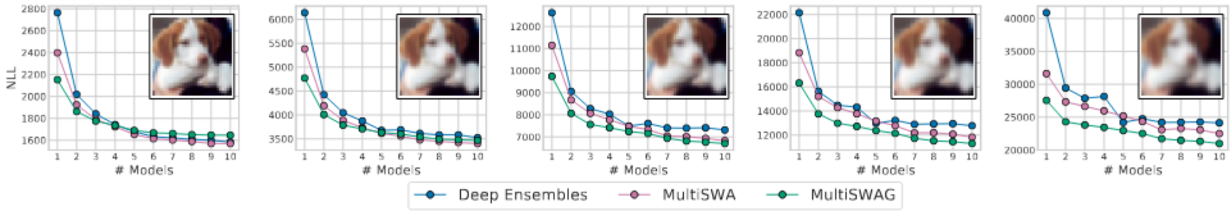


Figure 5. Negative log likelihood for Deep Ensembles, MultiSWAG and MultiSWA using a PreResNet-20 on CIFAR-10 with varying intensity of the *Gaussian blur* corruption. The image in each plot shows the intensity of corruption. For all levels of intensity, MultiSWAG and MultiSWA outperform Deep Ensembles for a small number of independent models. For high levels of corruption MultiSWAG significantly outperforms other methods even for many independent models. We present results for other corruptions in the Appendix.

fits of marginalization in deep learning, we need to consider as many hypotheses as possible through multimodal posterior approximations, such as MultiSWAG. In Section 7 we return to MultiSWAG, showing how it can entirely alleviate prominent double descent behaviour, and lead to striking improvements in generalization over SGD and single basin marginalization, for both accuracy and NLL.

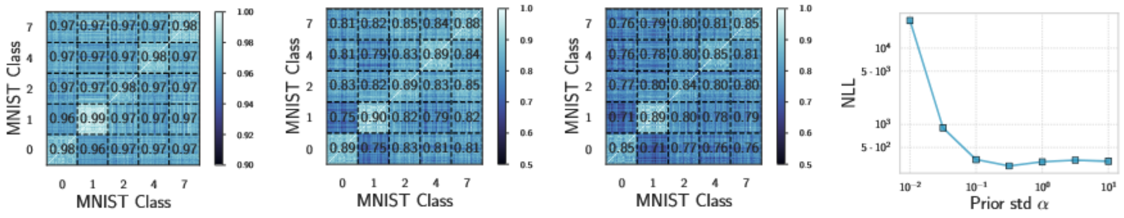
5. Neural Network Priors

A prior over parameters $p(w)$ combines with the functional form of a model $f(x; w)$ to induce a distribution over functions $p(f(x; w))$. It is this distribution over functions that controls the generalization properties of the model; the prior over parameters, in isolation, has no meaning. Neural networks are imbued with structural properties that provide good inductive biases, such as translation equivariance, hierarchical representations, and sparsity. In the sense of Figure 2, the prior will have large support, due to the flexibility of neural networks, but its inductive biases provide the most mass to datasets which are representative of problem settings where neural networks are often applied. In this section, we study the properties of the induced distribution over functions. We directly continue the discussion of priors in Section 6, with a focus on examining the noisy CIFAR results in Zhang et al. (2016), from a probabilistic perspective of generalization. These sections are best read together.

We also provide several additional experiments in the Appendix. In Section E, we present analytic results on the dependence of the prior distribution in function space on the variance of the prior over parameters, considering also layer-wise parameter priors with ReLU activations. As part of a discussion on tempering, in Section 8.4 we study the effect of α in $p(w) = \mathcal{N}(0, \alpha^2 I)$ on prior class probabilities for individual sample functions $p(f(x; w))$, the predictive distribution, and posterior samples as we observe varying amounts of data. In Section F, we further study the correlation structure over images induced by neural network priors, subject to perturbations of the images. In Section D.3 we provide additional experimental details.

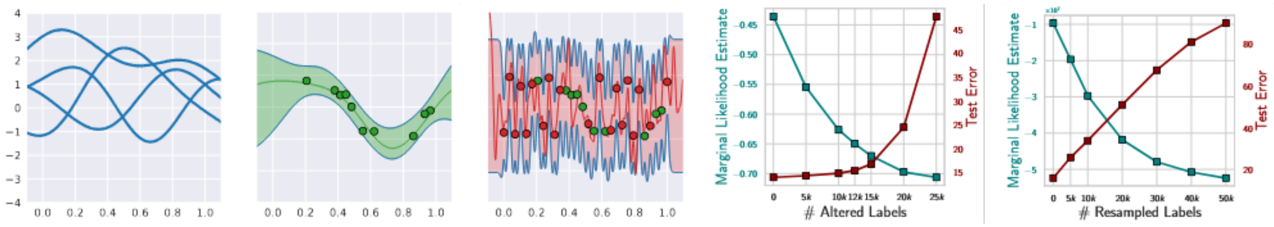
5.1. Deep Image Prior and Random Network Features

Two recent results provide strong evidence that vague Gaussian priors over parameters, when combined with a neural network architecture, induce a distribution over func-



(a) $\alpha = 0.02$ (b) $\alpha = 0.1$ (c) $\alpha = 1$ (d)

Figure 6. **Induced prior correlation function.** Average pairwise prior correlations for pairs of objects in classes $\{0, 1, 2, 4, 7\}$ of MNIST induced by LeNet-5 for $p(f(x; w))$ when $p(w) = \mathcal{N}(0, \alpha^2 I)$. Images in the same class have higher prior correlations than images from different classes, suggesting that $p(f(x; w))$ has desirable inductive biases. The correlations slightly decrease with increases in α. **(d)**: NLL of an ensemble of 20 SWAG samples on MNIST as a function of α using a LeNet-5.



(a) Prior Draws (b) True Labels (c) Corrupted Labels (d) Gaussian Process (e) PreResNet-20

Figure 7. **Rethinking generalization.** **(a)**: Sample functions from a Gaussian process prior. **(b)**: GP fit (with 95% credible region) to structured data generated as $y_{\text{green}}(x) = \sin(x \cdot 2\pi) + \epsilon,\ \ \epsilon \sim \mathcal{N}(0, 0.2^2)$. **(c)**: GP fit, with no training error, after a significant addition of corrupted data in red, drawn from Uniform$[0.5, 1]$. **(d)**: Variational GP marginal likelihood with RBF kernel for two classes of CIFAR-10. **(e)**: Laplace BNN marginal likelihood for a PreResNet-20 on CIFAR-10 with different fractions of random labels. The marginal likelihood for both the GP and BNN decreases as we increase the level of corruption in the labels, suggesting reasonable inductive biases in the prior over functions. Moreover, both the GP and BNN have 100% training accuracy on images with fully corrupted labels.

tions with useful inductive biases. In the *deep image prior*, Ulyanov et al. (2018) show that *randomly initialized* convolutional neural networks *without training* provide excellent performance for image denoising, super-resolution, and inpainting. This result demonstrates the ability for a sample function from a random prior over neural networks $p(f(x; w))$ to capture low-level image statistics, before any training. Similarly, Zhang et al. (2016) shows that preprocessing CIFAR-10 with a *randomly initialized untrained* convolutional neural network dramatically improves the test performance of a simple Gaussian kernel on pixels from 54% accuracy to 71%. Adding ℓ_2 regularization only improves the accuracy by an additional 2%. These results again indicate that *broad* Gaussian priors over parameters induce reasonable priors over networks, with a minor additional gain from decreasing the variance of the prior in parameter space, which corresponds to ℓ_2 regularization.

5.2. Prior Class Correlations

In Figure 6 we study the prior correlations in the outputs of the LeNet-5 convolutional network (LeCun et al., 1998) on objects of different MNIST classes. We sample networks with weights $p(w) = \mathcal{N}(0, \alpha^2 I)$, and compute the values of logits corresponding to the first class for all pairs of images and compute correlations of these logits. For all levels of α the correlations between objects corresponding to the same class are consistently higher than the correlation between objects of different classes, showing that the network induces a reasonable prior similarity metric over these images. Additionally, we observe that the prior correlations somewhat decrease as we increase α, showing that bounding the norm of the weights has some minor utility, in accordance with Section 5.1. Similarly, in panel (d) we see that the NLL significantly decreases as α increases in $[0, 0.5]$, and then slightly increases, but is relatively constant thereafter.

In the Appendix, we further describe analytic results and illustrate the effect of α on sample functions.

5.3. Effect of Prior Variance on CIFAR-10

We further study the effect of the parameter prior standard deviation α, measuring performance of approximate Bayesian inference for CIFAR-10 with a Preactivation ResNet-20 (He et al., 2016) and VGG-16 (Simonyan & Zisserman, 2014). For each of these architectures we run

SWAG (Maddox et al., 2019) with fixed hyper-parameters and varying α. We report the results in Figure 12(d), (h). For both architectures, the performance is near-optimal in the range $\alpha \in [10^{-2}, 10^{-1}]$. Smaller α constrains the weights too much. Performance is reasonable and becomes mostly insensitive to α as it continues to increase, due to the inductive biases of the functional form of the neural network.

6. Rethinking Generalization

Zhang et al. (2016) demonstrated that deep neural networks have sufficient capacity to fit randomized labels on popular image classification tasks, and suggest this result requires re-thinking generalization to understand deep learning.

We argue, however, that this behaviour is not puzzling from a probabilistic perspective, is not unique to neural networks, and cannot be used as evidence against Bayesian neural networks (BNNs) with vague parameter priors. Fundamentally, the resolution is the view presented in the introduction: from a probabilistic perspective, generalization is at least a *two-dimensional* concept, related to support (flexibility), which should be as large as possible, supporting even noisy solutions, and inductive biases that represent relative prior probabilities of solutions.

Indeed, we demonstrate that the behaviour in Zhang et al. (2016) that was treated as mysterious and specific to neural networks can be exactly reproduced by Gaussian processes (GPs). Gaussian processes are an ideal choice for this experiment, because they are popular Bayesian non-parametric models, and they assign a prior directly in function space. Moreover, GPs have remarkable flexibility, providing universal approximation with popular covariance functions such as the RBF kernel. Yet the functions that are a priori *likely* under a GP with an RBF kernel are relatively simple. We describe GPs further in the Appendix, and Rasmussen & Williams (2006) provides an extensive introduction.

We start with a simple example to illustrate the ability for a GP with an RBF kernel to easily fit a corrupted dataset, yet generalize well on a non-corrupted dataset, in Figure 7. In Fig 7(a), we have sample functions from a GP prior over functions $p(f(x))$, showing that likely functions under the prior are smooth and well-behaved. In Fig 7(b) we see the GP is able to reasonably fit data from a structured function. And in Fig 7(c) the GP is also able to fit highly corrupted data, with essentially no structure; although these data are not a likely draw from the prior, the GP has support for a wide range of solutions, including noise.

We next show that GPs can replicate the generalization behaviour described in Zhang et al. (2016) (experimental details in the Appendix). When applied to CIFAR-10 images with random labels, *Gaussian processes achieve 100% train accuracy*, and 10.4% test accuracy (at the level of random guessing). However, the same model trained on the true labels achieves a training accuracy of 72.8% and a test accuracy of 54.3%. Thus, the generalization behaviour described in Zhang et al. (2016) is not unique to neural networks, and can be described by separately understanding the support and the inductive biases of a model.

Indeed, although Gaussian processes support CIFAR-10 images with random labels, they are not likely under the GP prior. In Fig 7(d), we compute the approximate GP marginal likelihood on a binary CIFAR-10 classification problem, with labels of varying levels of corruption. We see as the noise in the data increases, the approximate marginal likelihood, and thus the prior support for these data, decreases. In Fig 7(e), we see a similar trend for a Bayesian neural network. Again, as the fraction of corrupted labels increases, the approximate marginal likelihood decreases, showing that the prior over functions given by the Bayesian neural network has less support for these noisy datasets. We provide further experimental details in the Appendix.

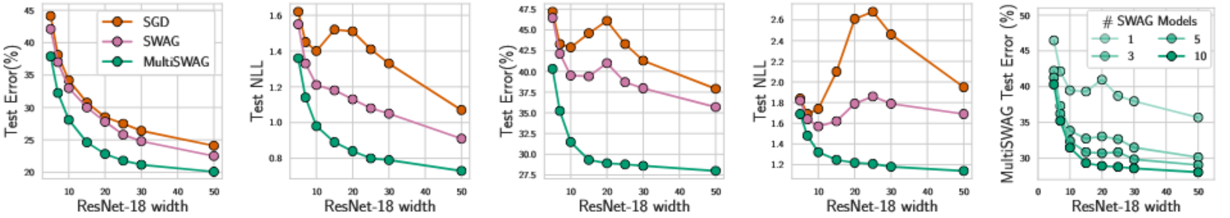
Dziugaite & Roy (2017) and Smith & Le (2018) provide complementary perspectives on Zhang et al. (2016), for MNIST; Dziugaite & Roy (2017) show non-vacuous PAC-Bayes bounds for the noise-free binary MNIST but not noisy MNIST, and Smith & Le (2018) show that logistic regression can fit noisy labels on subsampled MNIST, interpreting the results from an Occam factor perspective.

7. Double Descent

Double descent (e.g., Belkin et al., 2019) describes generalization error that decreases, increases, and then again decreases, with increases in model flexibility. The first decrease and then increase is referred to as the *classical regime*: models with increasing flexibility are increasingly able to capture structure and perform better, until they begin to overfit. The next regime is referred to as the *modern interpolating regime*. The existence of the interpolation regime has been presented as mysterious generalization behaviour in deep learning.

However, our perspective of generalization suggests that performance should monotonically improve as we increase model flexibility when we use Bayesian model averaging with a reasonable prior. Indeed, in the opening example of Figure 1, we would in principle want to use the most flexible possible model. Our results in Section 5 show that standard BNN priors induce structured and useful priors in the function space, so we should not expect double descent in Bayesian deep learning models that perform reasonable marginalization.

To test this hypothesis, we evaluate MultiSWAG, SWAG and standard SGD with ResNet-18 models of varying width, following Nakkiran et al. (2019), measuring both error and



(a) True Labels (Err) (b) True Labels (NLL) (c) Corrupted (Err) (d) Corrupted (NLL) (e) Corrupted (# Models)

Figure 8. **Bayesian model averaging alleviates double descent.** **(a)**: Test error and **(b)**: NLL loss for ResNet-18 with varying width on CIFAR-100 for SGD, SWAG and MultiSWAG. **(c)**: Test error and **(d)**: NLL loss when 20% of the labels are randomly reshuffled. SWAG reduces double descent, and MultiSWAG, which marginalizes over multiple modes, entirely alleviates double descent both on the original labels and under label noise, both in accuracy and NLL. **(e)**: Test errors for MultiSWAG with varying number of independent SWAG models; error monotonically decreases with increased number of independent models, alleviating double descent. We also note that MultiSWAG provides significant improvements in accuracy and NLL over SGD and SWAG models. See Appendix Figure 13 for additional results.

negative log likelihood (NLL). For the details, see Appendix D. We present the results in Figure 8 and Appendix Figure 13.

First, we observe that models trained with SGD indeed suffer from double descent, especially when the train labels are partially corrupted (see panels 8(c), 8(d)). We also see that SWAG, a unimodal posterior approximation, reduces the extent of double descent. Moreover, MultiSWAG, which performs a more exhaustive *multimodal* Bayesian model average *completely mitigates double descent*: the performance of MultiSWAG solutions increases monotonically with the size of the model, showing no double descent even under significant label corruption, for both accuracy and NLL. We also found that deep ensembles follow a similar pattern to MultiSWAG in Figure 8(c), also mitigating double descent, with slightly worse accuracy (about 1-2%). This result is in line with our perspective of Section 3.3 of deep ensembles providing a better approximation to the Bayesian predictive distribution than conventional single-basin Bayesian marginalization procedures.

Our results highlight the importance of marginalization over multiple modes of the posterior: under 20% label corruption SWAG clearly suffers from double descent while MultiSWAG does not. In Figure 8(e) we show how the double descent is alleviated with increased number of independent modes marginalized in MultiSWAG.

These results also clearly show that MultiSWAG provides significant improvements in *accuracy* over both SGD and SWAG models, in addition to NLL, an often overlooked advantage of Bayesian model averaging we discuss in Section 3.1.

Recently, Nakkiran et al. (2020) show that carefully tuned l_2 regularization can help mitigate double descent. Alternatively, we show that Bayesian model averaging, particularly based on multimodal marginalization, can mitigate prominent double descent behaviour. The perspective in Sections 1 and 3 predicts this result: models with reasonable priors and effective Bayesian model averaging should monotonically improve with increases in flexibility.

8. Temperature Scaling

The standard Bayesian posterior distribution is given by

$$p(w|\mathcal{D}) = \frac{1}{Z} p(\mathcal{D}|w)p(w), \tag{2}$$

where $p(\mathcal{D}|w)$ is a likelihood, $p(w)$ is a prior, and Z is a normalizing constant.

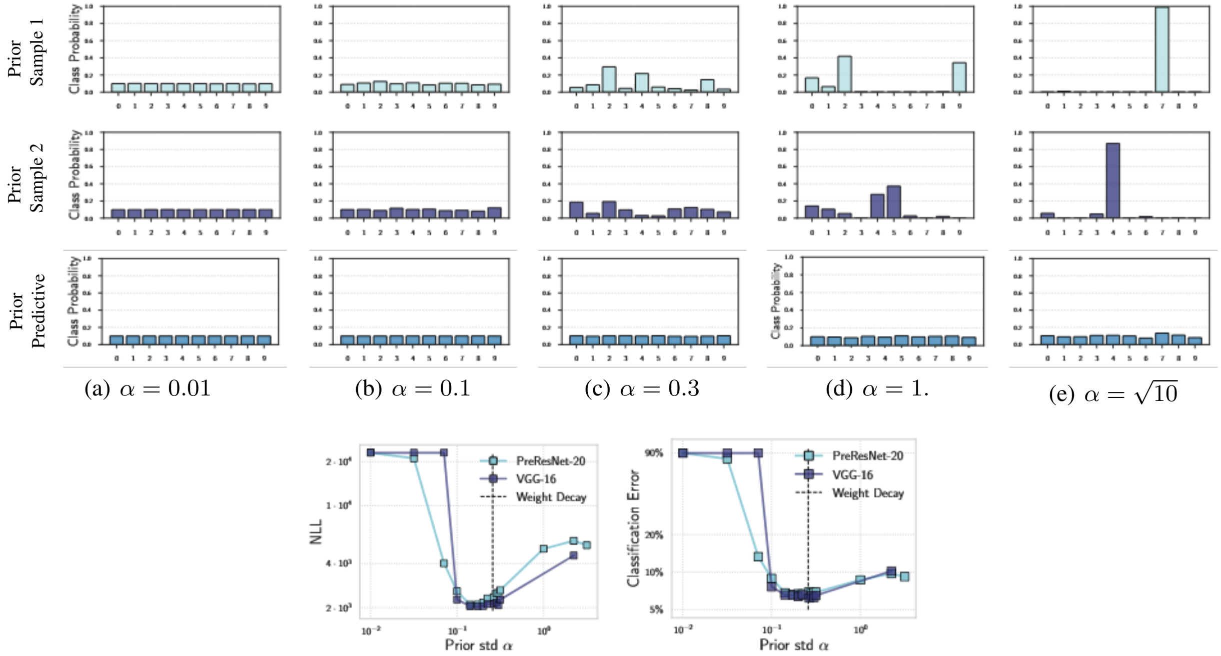
In Bayesian deep learning it is typical to consider the *tempered* posterior

$$p_T(w|\mathcal{D}) = \frac{1}{Z(T)} p(\mathcal{D}|w)^{1/T} p(w), \tag{3}$$

where T is a *temperature* parameter, and $Z(T)$ is the normalizing constant corresponding to temperature T. The temperature parameter controls how the prior and likelihood interact in the posterior:

- $T < 1$ corresponds to *cold posteriors*, where the posterior distribution is more concentrated around solutions with high likelihood.
- $T = 1$ corresponds to the standard Bayesian posterior distribution.
- $T > 1$ corresponds to *warm posteriors*, where the prior effect is stronger and the posterior collapse is slower.

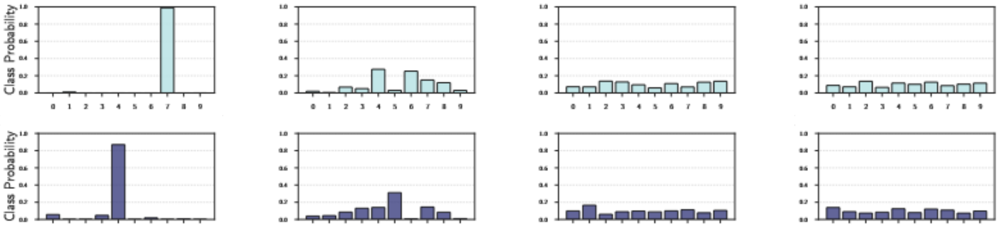
Tempering posteriors is a well-known practice in statistics, where it goes by the names *Safe Bayes*, *generalized*



(a) $\alpha = 0.01$ (b) $\alpha = 0.1$ (c) $\alpha = 0.3$ (d) $\alpha = 1.$ (e) $\alpha = \sqrt{10}$

(f) (g)

Figure 9. **Effects of the prior variance α^2. (a)–(e)**: Average class probabilities over all of CIFAR-10 for two sample prior functions $p(f(x; w))$ (two top rows) and predictive distribution (average over 200 samples of weights, bottom row) for varying settings of α in $p(w) = \mathcal{N}(0, \alpha^2 I)$. **(f)**: NLL and **(g)** classification error of an ensemble of 20 SWAG samples on CIFAR-10 as a function of α using a Preactivation ResNet-20 and VGG-16. The NLL is high for overly small α and near-optimal in the range of $[0.1, 0.3]$. The NLL remains relatively low for vague priors corresponding to large values of α.



(a) Prior ($\alpha = \sqrt{10}$) (b) 10 datapoints (c) 100 datapoints (d) 1000 datapoints

Figure 10. **Adaptivity of posterior variance with data**. We sample two functions $f(x; w)$ from the distribution over functions induced by a distribution over weights, starting with the prior $p(w) = \mathcal{N}(0, 10 \cdot I)$, in combination with a PreResNet-20. We measure class probabilities averaged across the CIFAR-10 test set, as we vary the amount of available training data. Although the prior variance is too large, such that the softmax saturates for logits sampled from the prior, leading to one class being favoured, we see that the posterior quickly adapts to correct the scale of the logits in the presence of data. In Figure 9 we also show that the prior variance can easily be calibrated such that the prior predictive distribution, even before observing data, is high entropy.

Bayesian inference, and *fractional Bayesian inference* (e.g., de Heide et al., 2019; Grünwald et al., 2017; Barron & Cover, 1991; Walker & Hjort, 2001; Zhang, 2006; Bissiri et al., 2016; Grünwald, 2012). Safe Bayes has been shown to be natural from a variety of perspectives, including from prequential, learning theory, and minimum description length frameworks (e.g., Grünwald et al., 2017).

Concurrently with our work, Wenzel et al. (2020) noticed that successful Bayesian deep learning methods tend to use cold posteriors. They provide an empirical study that shows that Bayesian neural networks (BNNs) with cold posteriors outperform models with SGD based maximum likelihood training, while BNNs with $T = 1$ can perform worse than the maximum likelihood solution. They claim that cold posteriors sharply deviate from the Bayesian paradigm, and consider possible reasons for why tempering is helpful in Bayesian deep learning.

In this section, we provide an alternative view and argue that tempering is not at odds with Bayesian principles. Moreover, for virtually any realistic model class and dataset, it would be highly surprising if $T = 1$ *were* in fact the best setting of this hyperparameter. Indeed, as long as it is practically convenient, we would advocate tempering for essentially *any* model, especially parametric models that do not scale their capacity automatically with the amount of available information. Our position is that at a high level Bayesian methods are trying to combine honest beliefs with data to form a posterior. By reflecting the belief that the model is misspecified, the tempered posterior is often more of a *true posterior* than the posterior that results from ignoring our belief that the model misspecified.

Finding that $T < 1$ helps for Bayesian neural networks is neither surprising nor discouraging. And the actual results of the experiments in Wenzel et al. (2020), which show great improvements over standard SGD training, are in fact very encouraging of deriving inspiration from Bayesian procedures in deep learning.

We consider (1) tempering under misspecification (Section 8.1); (2) tempering in terms of overcounting data (Section 8.2); (3) how tempering compares to changing the observation model (Section 8.3); (4) the effect of the prior in relation to the experiments of Wenzel et al. (2020) (Section 8.4); (5) the effect of approximate inference, including how tempering can help in efficiently estimating parameters even for the untempered posterior (Section 8.5).

This section shows how tempering can be a reasonable procedure, and addresses several of the points in Wenzel et al. (2020), particularly on prior misspecification.

Since the original publication of our paper, there have been many papers discussing the cold posterior effect. In our follow-up work (Izmailov et al., 2021), we show that there is no cold posterior effect in any of the examples of Wenzel et al. (2020) if we remove data augmentation. In Kapoor et al. (2022), we show precisely how data augmentation leads to underconfidence in Bayesian classification, and how posterior tempering can more naturally reflect our beliefs about aleatoric uncertainty than using $T = 1$. We also show that the cold posterior effect can be removed in the presence of data augmentation by using a Dirichlet observation model, which explicitly enables one to represent aleatoric uncertainty.

8.1. Tempering Helps with Misspecified Models

Many works explain how tempered posteriors help under model misspecification (e.g., de Heide et al., 2019; Grünwald et al., 2017; Barron & Cover, 1991; Walker & Hjort, 2001; Zhang, 2006; Bissiri et al., 2016; Grünwald, 2012). In fact, de Heide et al. (2019) and Grünwald et al. (2017) provide several simple examples where Bayesian inference fails to provide good convergence behaviour for untempered posteriors. While it is easier to show theoretical results for $T > 1$, several of these works also show that $T < 1$ can be preferred, even in well-specified settings, and indeed recommend learning T from data, for example by cross-validation (e.g., Grünwald, 2012; de Heide et al., 2019).

Are we in a misspecified setting for Bayesian neural networks? Of course. And it would be irrational to proceed as if it were otherwise. Every model is misspecified. In the context of Bayesian neural networks specifically, the mass of solutions expressed by the prior outside of the datasets we typically consider is likely much larger than desired for most applications. We can calibrate for this discrepancy through tempering. The resulting tempered posterior will be more in line with our beliefs than pretending the model is not misspecified and finding the untempered posterior.

Non-parametric models, such as Gaussian processes, attempt to side-step model misspecification by growing the number of free parameters (information capacity) automatically with the amount of available data. In parametric models, we take much more of a manual guess about the model capacity. In the case of deep neural networks, this choice is not even close to a *best guess*; it was once the case that architectural design was a large component of works involving neural networks, but now it is more standard practice to choose an off-the-shelf architecture, without much consideration of model capacity. We do not believe that knowingly using a misspecified model to find a posterior is more reasonable (or Bayesian) than honestly reflecting the belief that the model is misspecified and then using a tempered posterior. For parametric models such as neural networks, it is to be expected that the capacity is particularly misspecified.

8.2. Overcounting Data with Cold Posteriors

The criticism of cold posteriors raised by Wenzel et al. (2020) is largely based on the fact that decreasing temperature leads to overcounting data in the posterior distribution.

However, a similar argument can be made against marginal likelihood maximization (also known as *empirical Bayes* or *type 2 maximum likelihood*). Indeed, here, the prior will depend on the same data as the likelihood, which can lead to miscalibrated predictive distributions (Darnieder, 2011).

Nonetheless, empirical Bayes has been embraced and widely adopted in Bayesian machine learning (e.g., Bishop, 2006; Rasmussen & Williams, 2006; MacKay, 2003; Minka, 2001b), as embodying several Bayesian principles. Empirical Bayes has been particularly embraced in seminal work on Bayesian neural networks (e.g., MacKay, 1992; 1995), where it has been proposed as a principled approach to learning hyperparameters, such as the scale of the variance for the prior over weights, automatically embodying Occam's razor. While there is in this case some deviation from the fully Bayesian paradigm, the procedure, which depends on marginalization, is nonetheless clearly inspired by Bayesian thinking — and it is thus helpful to reflect this inspiration and provide understanding of how it works from a Bayesian perspective.

There is also work showing the marginal likelihood can lead to miscalibrated Bayes factors under model misspecification. Attempts to calibrate these factors (Xu et al., 2019), as part of the Bayesian paradigm, are highly reminiscent of work on safe Bayes.

8.3. Tempered Posterior or Different Likelihood?

In some cases, the tempered posterior for one model is an untempered posterior using a different likelihood function. Specifically, consider regression with a Gaussian likelihood and noise variance σ^2:

$$\begin{aligned} p(y|x,w) &= \mathcal{N}(y|f(x,w),\sigma^2) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{(y-f(x,w))^2}{2\sigma^2}\right), \end{aligned}$$

where $f(x,w)$ is the prediction of the model f with parameters w on the input x. Then, tempering the likelihood, we achieve

$$\begin{aligned} p(y|x,w)^{1/T} &= \frac{1}{\sqrt{2\pi\sigma^2}^{1/T}} \cdot \exp\left(-\frac{(y-f(x,w))^2}{2T\sigma^2}\right) \\ &= \mathcal{N}(y|f(x,w),T\sigma^2) \cdot \sqrt{\frac{(2\pi T\sigma^2)}{(2\pi\sigma^2)^{1/T}}} \\ &= \mathcal{N}(y|f(x,w),T\sigma^2) \cdot C, \end{aligned}$$

where C is a renormalization constant that does not depend on the parameters w of the model. In this case, the standard Bayesian posterior in the model with noise variance $T\sigma^2$ is equal to the posterior of temperature T in the original model with noise variance σ^2. Section 4.1 of Grünwald et al. (2017) considers a related construction.

The predictive distribution differs for the two models; even though the posteriors coincide, the likelihoods for a new datapoint y^* are different:

$$\int p(y^*|w)p(w)dw \neq \int p_T(y^*|w)p(w)dw\,. \qquad (4)$$

For the Gaussian model described above, the predictions of the tempered model and model with modified likelihood will have the same mean but different predictive variance.

Wenzel et al. (2020) provide a construction of a likelihood function that is equivalent to tempering for classification problems. For a general likelihood function $p(\mathcal{D}|w)$, we can consider a modified likelihood of the form

$$\hat{p}(\mathcal{D}|w) = p(\mathcal{D}|w)^{1/T} \cdot C(w), \qquad (5)$$

where $C(w)$ is a renormalization constant that in general depends on the parameters w and inputs x, but not the target values y. The standard posterior $\hat{p}(w|\mathcal{D})$ in the model with the modified likelihood will then coincide with the tempered posterior $p_T(w|\mathcal{D})$ in the original model up to $C(w)$:

$$\frac{\hat{p}(w|\mathcal{D})}{p_T(w|\mathcal{D})} = C(w). \qquad (6)$$

This subsection has been updated to add a discussion of renormalization, resolving a minor technical point. We however disagree with the discussion in Wenzel et al. (2020) on the connection between tempered posteriors and different likelihoods, and believe it misses the point. First, in many cases the tempered posterior can be exactly recovered by changing the likelihood on the training data, such as for regression with Gaussian noise. Moreover, as above, we can simply introduce a renormalization constant that does not depend on the labels y, to preserve the equivalence of a tempered posterior and a modified likelihood in any model. This renormalization constant can be viewed as a prior over the parameters that we implicitly define with respect to the predictions on the training data; such a prior can, for example, encode beliefs about how confident the network should be in its predictions on the training data.

Furthermore, as Wenzel et al. (2020) show, the tempered softmax likelihood with $T < 1$ can be viewed as a valid likelihood if we introduce a new class, not observed in the training data. While they discard that particular interpretation, it is not unreasonable to include an unobserved class, since our observation model may want to recognize that we have not observed all possible classes, and therefore retain an additional label. This extra class can, for example, correspond to all the possible images that do not belong to any of the classes in our dataset. Finally, new work (Kapoor et al., 2022) shows that we can naturally interpret the tempered likelihood as using the multinomial observation model, assuming $1/T$ counts of the label are observed for each of the training datapoints, which is perfectly valid.

We present other key considerations in the discussion of tempering for Bayes posteriors in other parts of this section.

8.4. Effect of the Prior

While a somewhat misspecified prior will certainly interact with the utility of tempering, we do not believe the experiments in Wenzel et al. (2020) provide evidence that even the prior $p(w) = \mathcal{N}(0, I)$ is misspecified to any serious extent. For a relatively wide range of distributions over w, the functional form of the network $f(x; w)$ can produce a generally reasonable distribution over functions $p(f(x; w))$. In Figure 10, we reproduce the findings in Wenzel et al. (2020) showing sample functions $p(f(x; w))$ corresponding to the prior $p(w) = \mathcal{N}(0, 10 \cdot I)$ strongly favour a single class over the dataset. While this behaviour appears superficially dramatic, we note it is simply an artifact a miscalibrated signal variance. A miscalibrated signal variance interacts with a quickly saturating soft-max link function to provide a seemingly dramatic preference to a given class. If we instead use $p(w) = \mathcal{N}(0, \alpha^2 I)$, for quite a range of α, then sample functions provide reasonably high entropy across labels averaged over the dataset, as in Figure 9. For individual points the posterior samples have different particular class preferences. α can be easily determined through cross-validation, as in Figure 9, or specified as a standard value used for L_2 regularization ($\alpha = 0.24$ in this case).

However, even with the inappropriate prior scale, we see in panels (a)–(e) of Figure 9 that the unconditional predictive distribution *is* completely reasonable. Moreover, the prior variance represents a *soft* prior bias, and will quickly update with data. In Figure 10 we show posterior samples after observing $10, 100$, and 1000 data points.

Other aspects of the prior, outside of the prior signal variance, will have a much greater effect on the inductive biases of the model. For example, the induced covariance function $\text{cov}(f(x_i, w), f(x_j, w))$ reflects the induced similarity metric over data instances; through the covariance function we can answer, for instance, whether the model believes a priori that a translated image is similar to the original. Unlike the signal variance of the prior, the prior covariance function will continue to have a significant effect on posterior inference for even very large datasets, and strongly reflects the structural properties of the neural network. We explore these structures of the prior in Figure 12.

8.5. The Effect of Inexact Inference

We have to keep in mind what we ultimately use posterior samples to compute. Ultimately, we wish to estimate the predictive distribution given by the integral in Equation (1). With a finite number of samples, the tempered posterior could be used to provide a better approximation to the expectation of the predictive distribution associated with untempered posterior.

Consider a simple example, where we wish to estimate the mean of a high-dimensional Gaussian distribution $\mathcal{N}(0, I)$. Suppose we use J independent samples. The mean of these samples is also Gaussian distributed, $\mu \sim \mathcal{N}(0, \frac{1}{J} I)$. In Bayesian deep learning, the dimension d is typically on the order 10^7, and J would be on the order of 10. The norm of μ would be highly concentrated around $\frac{\sqrt{10^7}}{\sqrt{10}} = 1000$. In this case, sampling from a tempered posterior with $T < 1$ would lead to a better approximation of the Bayesian model average associated with an untempered posterior.

Furthermore, no current sampling procedure will be providing samples that are close to independent samples from the true posterior of a Bayesian neural network. The posterior landscape is far too multimodal and complex for there to be any reasonable coverage. The approximations we have are practically useful, and often preferable to conventional training, but we cannot realistically proceed with analysis assuming that we have obtained true samples from a posterior. While we would expect that some value of $T \neq 1$ would be preferred for any finite dataset in practice, it is conceivable that some of the results in Wenzel et al. (2020) may be affected by the specifics of the approximate inference technique being used.

We should be wary not to view Bayesian model averaging purely through the prism of simple Monte Carlo, as advised in Section 3.2. Given a finite computational budget, our goal in effectively approximating a Bayesian model average is *not* equivalent to obtaining good samples from the posterior.

9. Discussion

> *"It is now common practice for Bayesians to fit models that have more parameters than the number of data points… Incorporate every imaginable possibility into the model space: for example, if it is conceivable that a very simple model might be able to explain the data, one should include simple models; if the noise might have a long-tailed distribution, one should include a hyperparameter which controls the heaviness of the tails of the distribution; if an input variable might be irrelevant to a regression, include it in the regression anyway."* MacKay (1995)

We have presented a probabilistic perspective of generalization, which depends on the support and inductive biases of the model. The support should be as large possible, but the inductive biases must be well-calibrated to a given problem class. We argue that Bayesian neural networks embody these properties — and through the lens of probabilistic inference, explain generalization behaviour that has previously been viewed as mysterious. Moreover, we argue that Bayesian marginalization is particularly compelling for neural networks, show how deep ensembles provide a practical mechanism for marginalization, and propose a new approach that generalizes deep ensembles to marginalize within basins of attraction. We show that this multimodal approach to Bayesian model averaging, MultiSWAG, can entirely alleviate double descent, to enable monotonic performance improvements with increases in model flexibility, as well significant improvements in generalization accuracy and log likelihood over SGD and single basin marginalization.

There are certainly many challenges to estimating the integral for a Bayesian model average in modern deep learning, including a high-dimensional parameter space, and a complex posterior landscape. But viewing the challenge indeed as an integration problem, rather than an attempt to obtain posterior samples for a simple Monte Carlo approximation, provides opportunities for future progress. Bayesian deep learning has been making fast practical advances, with approaches that now enable better accuracy and calibration over standard training, with minimal overhead.

We finish with remarks about future developments for Bayesian neural network priors, and approaches to research in Bayesian deep learning.

9.1. The Future for BNN Priors

We provide some brief remarks about future developments for BNN priors. Here we have explored relatively simple parameter priors $p(w) = \mathcal{N}(0, \alpha^2 I)$. While these priors are simple in parameter space, they interact with the neural network architecture to induce a sophisticated prior over functions $p(f(x; w))$, with many desirable properties, including a reasonable correlation structure over images. However, these parameter priors can certainly still be improved. As we have seen, even tuning the value of the signal variance α^2, an analogue of the L_2 regularization often used in deep learning, can have a noticeable affect on the induced prior over functions — though this affect is quickly modulated by data. Layer-wise priors, such that parameters in each layer have a different signal variance, are intuitive: we would expect later layers require precise determination, while parameters in earlier layers could reasonably take a range of values. But one has to be cautious; as we show in Appendix Section E, with ReLU activations different signal variances in different layers can be degenerate, combining together to affect only the output scale of the network.

A currently popular sentiment is that we should directly build function-space BNN priors, often taking inspiration from Gaussian processes. While we believe this is a promising direction, one should proceed with caution. If we contrive priors over parameters $p(w)$ to induce distributions over functions $p(f)$ that resemble familiar models such as Gaussian processes with RBF kernels, we could be throwing the baby out with the bathwater. Neural networks are useful as their own model class precisely because they have different inductive biases from other models.

A similar concern applies to taking infinite width limits in Bayesian neural networks. In these cases we recover Gaussian processes with interpretable kernel functions; because these models are easier to use and analyze, and give rise to interpretable and well-motivated priors, it is tempting to treat them as drop-in replacements for the parametric analogues. However, the kernels for these models are *fixed*. In order for a model to do effective representation learning, we must learn a similarity metric for the data. Training a neural network in many ways is like *learning* a kernel, rather than using a fixed kernel. MacKay (1998) has also expressed concerns in treating these limits as replacements for neural networks, due to the loss of representation learning power.

Perhaps the distribution over functions induced by a network in combination with a generic distribution over parameters $p(w)$ may be hard to interpret — but this distribution will contain the equivariance properties, representation learning abilities, and other biases that make neural networks a compelling model class in their own right.

9.2. "But is it *really* Bayesian?"

We finish with an editorial comment about approaches to research within Bayesian deep learning. There is sometimes a tendency to classify work as *Bayesian* or *not Bayesian*, with very stringent criteria for what qualifies as *Bayesian*. Moreover, the implication, and sometimes even explicit recommendation, is that if an approach is not unequivocally Bayesian in every respect, then we should not term it as Bayesian, and we should instead attempt to understand the procedure through entirely different non-Bayesian mechanisms. We believe this mentality encourages tribalism, which is not conducive to the best research, or creating the best performing methods. What matters is not a debate about semantics, but making rational modelling choices given a particular problem setting, and trying to understand these choices. Often these choices can largely be inspired by a Bayesian approach — in which case it desirable to indicate this source of inspiration. And in the semantics debate, who would be the arbiter of what gets to be called Bayesian? Arguably it ought to be an evolving definition.

Broadly speaking, what makes Bayesian approaches distinctive is a posterior weighted marginalization over parameters. And at a high level, Bayesian methods are about combining our honest beliefs with data to form a posterior. In actuality, no fair-minded researcher entirely believes the prior over parameters, the functional form of the model (which is part of the prior over functions), or the likelihood. From this perspective, it is broadly compatible with a Bayesian philosophy to reflect misspecification in the modelling procedure itself, which is achieved through tempering. In this sense, the *tempered posterior* is more reflective of a *true posterior* than the posterior that results from ignoring our belief that the model is misspecified.

Moreover, basic probability theory indicates that marginalization is desirable. While marginalization cannot in practice be achieved exactly, we can try to improve over conventional training, which as we have discussed can be viewed as approximate marginalization. Given computational constraints, effective marginalization is not equivalent to obtaining accurate samples from a posterior. As we have discussed, simple Monte Carlo is only one of many mechanisms for marginalization. Just like we how expectation propagation (Minka, 2001a) focuses its approximation to factors in a posterior where it will most affect the end result, we should focus on representing the posterior where it will make the biggest difference to the model average. As we have shown, deep ensembles are a reasonable mechanism up to a point. After having trained many independent models, there are added benefits to marginalizing within basins, given the computational expense associated with retraining an additional model to find an additional basin of attraction.

We should also not hold Bayesian methods to a double standard. Indeed, it can be hard to interpret or understand the prior, the posterior, and whether the marginalization procedure is optimal. But it is also hard to interpret the choices behind the functional form of the model, or the rationale behind classical procedures where we bet everything on a single global optimum — when we know there are many global optima and many of them will perform well but provide different solutions, and many others will not perform well. We should apply the same level of scrutiny to all modelling choices, consider the alternatives, and not be paralyzed if a procedure is not optimal in every respect.

Acknowledgements. AGW and PI are supported by an Amazon Research Award, Facebook Research, NSF I-DISRE 193471, NIH R01 DA048764-01A1, NSF IIS-1563887, and NSF IIS-1910266. We thank Greg Benton for helpful discussions.

References

Achille, A. and Soatto, S. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1):1947–1980, 2018.

Alquier, P. User-friendly introduction to pac-bayes bounds. *arXiv preprint arXiv:2110.11216*, 2021.

Ashukha, A., Lyzhov, A., Molchanov, D., and Vetrov, D. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *arXiv preprint arXiv:2002.06470*, 2020.

Barron, A. R. and Cover, T. M. Minimum complexity density estimation. *IEEE transactions on information theory*, 37(4):1034–1054, 1991.

Beal, M. J. *Variational algorithms for approximate Bayesian inference*. university of London, 2003.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

Bissiri, P. G., Holmes, C. C., and Walker, S. G. A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):1103–1130, 2016.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

Box, G. E. and Tiao, G. C. Bayesian inference in statistical analysis, addision-wesley. *Reading, MA*, 1973.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019.

Cobb, A. D., Baydin, A. G., Markham, A., and Roberts, S. J. Introducing an explicit symplectic integration scheme for riemannian manifold hamiltonian monte carlo. *arXiv preprint arXiv:1910.06243*, 2019.

Darnieder, W. F. *Bayesian methods for data-dependent priors*. PhD thesis, The Ohio State University, 2011.

de Heide, R., Kirichenko, A., Mehta, N., and Grünwald, P. Safe-Bayesian generalized linear regression. *arXiv preprint arXiv:1910.09227*, 2019.