3rd Symposium on Advances in Approximate Bayesian Inference, 2020 1–9

Sensible Priors for Bayesian Neural Networks through Wasserstein Distance Minimization to Gaussian Processes

Anonymous Authors
Anonymous Institution

Abstract

The Bayesian treatment of neural networks dictates that a prior distribution is considered over the weight and bias parameters of the network. The non-linear nature of the model implies that any distribution of the parameters has an unpredictable effect on the distribution of the function output. Gaussian processes offer a rigorous framework to define prior distributions over the space of functions. Our proposal is to impose such functional priors on well-established architectures of neural networks by means of minimising the Wasserstein distance between samples of stochastic processes. Early experimental results demonstrate the potential of functional priors for Bayesian neural networks.

1. Introduction

The concept of prior distribution in Bayesian inference offers a mathematical formulation that allows us to describe the family of solutions that we consider acceptable, *before* having seen any data. While there are cases for which picking a good prior is easy or intuitive given the context (O'Hagan, 1991; Ghahramani and Rasmussen, 2003; Srinivas et al., 2010; Cockayne et al., 2019; Briol et al., 2019), for nonlinear parametric models with thousands (or millions) parameters, like deep neural networks (DNNs) and convolutional neural networks (CNNs), this choice is not straightforward. As these models are nowadays accepted as *de facto standard* in machine learning (LeCun et al., 2015), the community has been actively proposing ways to reason about the uncertainty of their predictions, with the Bayesian machinery being at the core of many contributions (Graves, 2011; Chen et al., 2014; Gal and Ghahramani, 2016; Liu and Wang, 2016). Despite many advances in the field (Kendall and Gal, 2017; Rossi et al., 2019; Osawa et al., 2019; Rossi et al., 2020), it is reported that in some cases the predictive posteriors are not competitive to non-Bayesian alternatives, making these models—and Bayesian deep learning, in general—less than ideal solutions for a number of cases, if no remedies are taken (Wenzel et al., 2020).

In this work, we focus our discussion on the prior distribution of Bayesian neural networks (BNNs). For such models, the common practice is to define a prior distribution on the network weights and biases, often a Gaussian distribution. A prior on the parameters induces a prior on the functions generated by the model, which also depends on the network architecture. However, due to the nonlinear nature of the model, the effect of this prior on the functional output is not obvious to characterize and control.

Consider the example in Fig. 1, where we show the functions generated by sampling the weights of BNNs with tanh activation from their Gaussian prior $\mathcal{N}(0, 1)$. We see that as depth increases the samples tend to form straight horizontal lines with large marginal variance, which is a pathology known in the literature (Duvenaud et al., 2014). We stress that

© A. Authors.

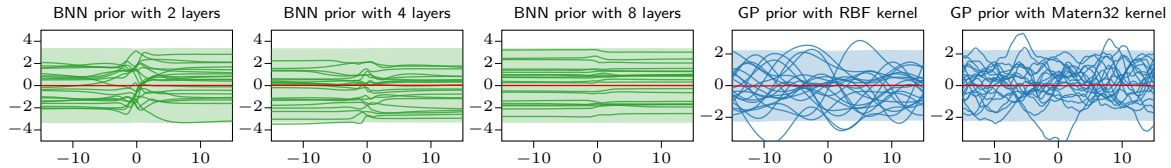

Figure 1: *Left:* Samples of a fully-connected BNN with 2, 4 and 8 layers with a Gaussian prior on the weights. *Right:* Samples from GP prior with two different kernels.

a fixed Gaussian prior on the parameters is not always problematic, but it can be, especially for deeper architectures. Nonetheless, this kind of generative priors on the functions is very different from shallow Bayesian models, such as Gaussian Processes (GPs), where the selection of an appropriate prior typically reflects certain attributes that we expect from the generated functions. A GP defines a distribution of functions that is characterized by a mean and a kernel function κ. The GP prior specification can be more *interpretable* than the one induced by the prior over the weights of a BNN, in the sense that the kernel effectively governs the properties of possible functions, such as shape, variability and smoothness.

Contributions We seek to tune BNNs so that their (functional) prior distributions exhibit interpretable properties, similar to shallow GPs. We consider the *Wasserstein distance* between the distribution of functions induced by BNN prior over the network parameters, and a target GP prior. We propose an algorithm that optimizes BNN hyper-parameters based on the distance between the BNN and a desired GP. To the best of our knowledge, our approach is novel and it leads to sensible improvements over classical choices for the prior.

Related works It is well-known that the prior over functions induced by a shallow BNN converges to a GP as it width becomes large (Neal, 1996; de G. Matthews et al., 2018). This motivates a line of research that attempts to map GP priors to BNN priors (Flam-Shepherd et al., 2017; Pearce et al., 2019). Closest to our work is that of Flam-Shepherd et al. (2017), which propose to minimise the Kullback-Leibler (KL) divergence between the BNN prior and a target GP prior. As there is no analytical form for this KL due the entropy term, the Authors propose to ignore this term and employ an early stopping scheme as it merely acts as a regularization term. We compare with the KL-based approach in the following sections.

2. Using the Wasserstein distance to map GP prior in Bayesian NN

The equivalence between function-space view and weight-space view of linear models, like Bayesian linear regression and GPs (Rasmussen and Williams, 2005), is a straightforward application of Gaussian identities, but it allows us to seamlessly switch point of view accordingly to which characteristics of the model we are willing to observe or impose. We would like to leverage this equivalence also for BNNs but the nonlinear nature of such models makes it analytically intractable (or impossible, for non-invertible activation functions). We argue that for BNNs—and Bayesian deep learning models, in general—starting from a prior over the weights is not ideal, given the impossibility of interpreting its effect on the family of functions that the model can represent. We therefore rely on an optimization-based procedure to impose functional priors on BNNs using the Wasserstein distance as a similarity metric between such distributions.

In the general case, given two probability measures $\pi(\mathbf{x})$ and $\nu(\mathbf{y})$ defined on a separable metric space $\mathcal{X}$ (e.g. $\mathbb{R}^d$), the 1–Wasserstein distance is defined as follows

$$W_1(\pi, \nu) = \inf_{\gamma \in \Gamma(\pi,\nu)} \int D(\mathbf{x}, \mathbf{y}) \gamma(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{y} \,, \tag{1}$$

where $D(\mathbf{x}, \mathbf{y})$ is a proper distance metric between two points $\mathbf{x}$ and $\mathbf{y}$ (e.g. Euclidean norm distance $D(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$) and $\Gamma(\pi, \nu)$ is the set of functionals of all possible joint densities γ whose marginals are π and ν. With the exception of few cases where the solution is available analytically (e.g. π and ν being Gaussians), solving Eq. 1 directly or via optimization is intractable. Fortunately, the Wasserstein distance in Eq. 1 admits the following dual form (Kantorovich, 1942, 1948),

$$W_1(\pi, \nu) = \sup_{\|\phi\|_L \leq 1} \mathbb{E}_{\mathbf{x} \sim \pi} \phi(\mathbf{x}) - \mathbb{E}_{\mathbf{y} \sim \nu} \phi(\mathbf{y}) \,, \tag{2}$$

where ϕ is a 1-Lipschitz continuous function defined on $\mathcal{X} \to \mathbb{R}$. This is effectively a functional maximization of ϕ on the difference between two expectations of ϕ, evaluated under π and ν, constrained to its Lipschitz continuity.

2.1. Wasserstein distance optimization for mapping priors

Assume to have a BNN with weights $\mathbf{w}$ on which we have a prior distribution $p(\mathbf{w}; \boldsymbol{\psi})$, where $\boldsymbol{\psi}$ is its set of parameters (e.g. for a Gaussian prior, $\boldsymbol{\psi} = \{\mu, \sigma^2\}$). This prior on the weights induces a corresponding prior distribution over functions $p_{\boldsymbol{nn}}(\mathbf{f}; \boldsymbol{\psi}) = \int p(\mathbf{f} \,|\, \mathbf{w}) p(\mathbf{w}; \boldsymbol{\psi}) \, \mathrm{d}\mathbf{w}$, where $p(\mathbf{f} \,|\, \mathbf{w})$ is deterministically defined by the network architecture. Our target GP prior is $p_{\boldsymbol{gp}}(\mathbf{f}) = \mathcal{N}(\mathbf{f} \,|\, \mathbf{0}, \mathbf{K})$, where $\mathbf{K}$ is the covariance matrix obtained by computing the kernel function κ for each pair of $\{\mathbf{x}_i, \mathbf{x}_j\}$ in the training set. We aim at matching these two stochastic processes at a finite number of measurement points $\mathbf{X}_{\mathcal{M}} \stackrel{\text{def}}{=} [\mathbf{x}_1, ..., \mathbf{x}_M]^\top$ sampled from a distribution $q(\mathbf{x})$. To achieve that, we propose a sample-based approach using the 1-Wasserstein distance (Eq. 2) as the objective

$$\min_{\boldsymbol{\psi}} \max_{\boldsymbol{\theta}} \mathbb{E}_q \Big[\underbrace{\mathbb{E}_{p_{\boldsymbol{gp}}} [\phi_{\boldsymbol{\theta}}(\mathbf{f}_{\mathcal{M}})] - \mathbb{E}_{p_{\boldsymbol{nn}}} [\phi_{\boldsymbol{\theta}}(\mathbf{f}_{\mathcal{M}})]}_{\mathcal{L}(\boldsymbol{\psi}, \boldsymbol{\theta})} \Big] \,, \tag{3}$$

where $\mathbf{f}_{\mathcal{M}}$ are the function values at $\mathbf{X}_{\mathcal{M}}$ and ϕ is 1-Lipschitz function. Following recent literature (Goodfellow et al., 2014), we parameterize the Lipschitz function ϕ by a neural network with parameters $\boldsymbol{\theta}$[1]. We alternate between $n_{\text{Lipschitz}}$ steps of maximising $\mathcal{L}$ with respect to the Lipschitz function's parameters $\boldsymbol{\theta}$ and one step of minimizing $\mathcal{L}$ with respect to the prior's parameters $\boldsymbol{\psi}$. Notice that the objective is fully sample-based. As a result, it is not necessary to know the closed-form of the marginal density $p_{\boldsymbol{nn}}(\mathbf{f}; \boldsymbol{\psi})$.

Lipschitz constraint. Based on the fact that a differentiable function is 1-Lipschitz if and only if the norm of its gradient is at most one everywhere, Gulrajani et al. (2017) propose a

1. Details on the 1-Lipschitz function: we used a multilayer perceptron (MLP) with two hidden layers, each with 200 units; the activation function is softplus, which is defined as: $\text{softplus}(x) = 1/(1 + \exp(-x))$.

soft constrain on the gradient norm of the output of the Lipschitz function $\phi_{\boldsymbol{\theta}}$ with respect to its input. The loss of the Lipschitz function is then augmented by a regularization term

$$\mathcal{L}_R(\boldsymbol{\psi}, \boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\psi}, \boldsymbol{\theta}) + \underbrace{\lambda \mathbb{E}_{p_{\hat{\mathbf{f}}}} \left[\left(\left\| \boldsymbol{\nabla}_{\phi} \hat{\mathbf{f}} \right\|_2 - 1 \right)^2 \right]}_{\text{Gradient penalty}}. \tag{4}$$

Here $\hat{\mathbf{f}}$ is the distribution of $\hat{\mathbf{f}} = \varepsilon \mathbf{f}_{nn} + (1-\varepsilon)\mathbf{f}_{gp}$ for $\varepsilon \sim \mathcal{U}[0,1]$ and $\mathbf{f}_{nn} \sim p_{nn}$, $\mathbf{f}_{gp} \sim p_{gp}$ being the sample functions from BNN and GP priors, respectively; λ is the penalty coefficient.

Choice of the measurement set. We are using finite measurement sets to have a practical and well-defined optimization. For low-dimensional problems, one can simply use a regular grid or apply uniform sampling in the input domain. For high-dimensional problems, one can sample from the training set or an augmented version of it, where noise is injected into the data. We follow a combination of the two approaches: we use the training inputs (or a subset of thereof) to which we include points that are randomly sampled (uniformly) in the input domain. A pseudocode of the procedure is formally presented in Algorithm 1.

Algorithm 1: Wasserstein Distance Optimization

Requires: B, mini-batch size; $q(\mathbf{x})$, sampling distribution for measurement set; $n_{\text{Lipschitz}}$, number of iterations of Lipschitz function per prior iteration;

while $\boldsymbol{\psi}$ *has not converged* **do**
- draw $\mathbf{X}_{\mathcal{M}}$ from $q(\mathbf{x})$ `// Sample measurement set` ;
- **for** $t = 1, ..., n_{\text{Lipschitz}}$ **do**
  - draw GP functions $\{\mathbf{f}_{gp}^{(i)}\}_{i=1}^{B} \sim p_{gp}(\mathbf{f}; \kappa)$ at $\mathbf{X}_{\mathcal{M}}$;
  - draw NN functions $\{\mathbf{f}_{nn}^{(i)}\}_{i=1}^{B} \sim p_{nn}(\mathbf{f}; \boldsymbol{\psi})$ at $\mathbf{X}_{\mathcal{M}}$;
  - $\mathcal{L}_R = B^{-1} \sum_{i=1}^{B} \mathcal{L}_R^{(i)}$ `// Compute Lipschitz objective` $\mathcal{L}_R$ `using Eq. 4` ;
  - $\boldsymbol{\theta} \leftarrow \text{Optimiser}(\boldsymbol{\theta}, \boldsymbol{\nabla}_{\boldsymbol{\theta}} \mathcal{L}_R)$ `// Update Lipschitz function` ϕ ;
- **end**
- draw GP functions $\{\mathbf{f}_{gp}^{(i)}\}_{i=1}^{B} \sim p_{gp}(\mathbf{f}; \kappa)$ at $\mathbf{X}_{\mathcal{M}}$;
- draw NN functions $\{\mathbf{f}_{nn}^{(i)}\}_{i=1}^{B} \sim p_{nn}(\mathbf{f}; \boldsymbol{\psi})$ at $\mathbf{X}_{\mathcal{M}}$;
- $\widetilde{W}_1 = B^{-1} \sum_{i=1}^{B} \phi_{\boldsymbol{\theta}}\left(\mathbf{f}_{gp}^{(i)}\right) - \phi_{\boldsymbol{\theta}}\left(\mathbf{f}_{nn}^{(i)}\right)$ `// Compute Wasserstein-1 distance using Eq. 3` ;
- $\boldsymbol{\psi} \leftarrow \text{Optimiser}(\boldsymbol{\psi}, \boldsymbol{\nabla}_{\boldsymbol{\psi}} \widetilde{W}_1)$ `// Update prior` p_{nn} ;

end

3. Experimental validation

Before presenting the experimental results on the classic UCI benchmark for regression tasks, we report the details on the choice of prior distributions and we illustrate the advantages offered by the use of the Wasserstein distance compared to alternative approaches.

3.1. Parametrization of BNN Prior

First, we observe that our proposal and Algorithm 1 is completely independent on the choice of the form of the prior on the weights and biases of BNNs. The question that now remains is *"What form of prior should we place on* $\mathbf{w}$*?"* Since the family of functions the BNN can

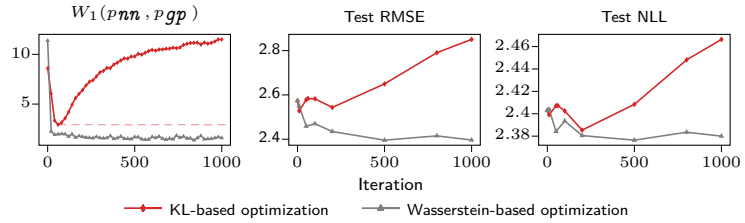

Figure 2: Comparison between KL-based and Wasserstein-based optimization

represent depends on both the architecture of the network and the activation function, we argue that there is not a unique choice.

We consider a layer-wise factorization with two independent Gaussian distributions for weights and biases. The parameters to adjust are $\boldsymbol{\psi} = \{\sigma^2_{l_\mathrm{w}}, \sigma^2_{l_\mathrm{b}}\}_{l=1}^{L}$, where $\sigma^2_{l_\mathrm{w}}$ is the prior variance shared across all weights in layer l, and $\sigma^2_{l_\mathrm{b}}$ is the respective variance for the bias parameters. For any weight and bias entries $w, b \in \mathbf{w}$, the prior is:

$$p(w) = \mathcal{N}\left(w;\ 0, \sigma^2_{l_\mathrm{w}}\right) \quad \text{and} \quad p(b) = \mathcal{N}\left(b;\ 0, \sigma^2_{l_\mathrm{b}}\right)$$

In the next Section, we refer to this parameterization as the GP induced Gaussian (GPiG) prior. Although this simple approach retains the Gaussian prior on the parameters, in many cases it is shown to sufficiently capture the target functional priors.

3.2. KL divergence or Wasserstein distance?

The KL divergence is popular choice as an optimization criterion capturing similarity between distributions. It was used in the same vein as in this work by Flam-Shepherd et al. (2017), where they consider the KL divergence between samples of a BNN and a GP. This requires an empirical estimation of the entropy, which is a challenging task for high-dimensional distributions (Delattre and Fournier, 2017). These issues were also reported by Flam-Shepherd et al. (2017), where they propose an early stopping scheme to what is essentially an optimization of the cross-entropy term.

In our experiments, we have found that a scheme based on Wasserstein distance converges more consistently without the need for additional heuristics. We demonstrate the converge properties of our scheme against Flam-Shepherd et al. (2017) in Fig. 2. We monitor the evolution of Wasserstein distance from the target GP prior and performance metrics for the Boston UCI dataset (test negative loglikelihood (NLL) and root mean square error (RMSE)). The KL-based scheme provides improvements for the first few iterations, and then early stopping is required, as prescribed by Flam-Shepherd et al. (2017) (dashed red line). Our approach, instead, improves over the iterations and does not need early stopping to achieve converge to the desired prior, and it offers consistently better performance.

3.3. Regression experiments on the UCI benchmark

We evaluate our proposed method on a classic UCI benchmark for regression. We consider two fixed priors: (1) the Fixed Gaussian prior (FG), $\mathcal{N}(0, 1)$; (2) the Fixed Hierarchical prior (FH) where the prior variance for each layer is sampled from an Inverse-Gamma

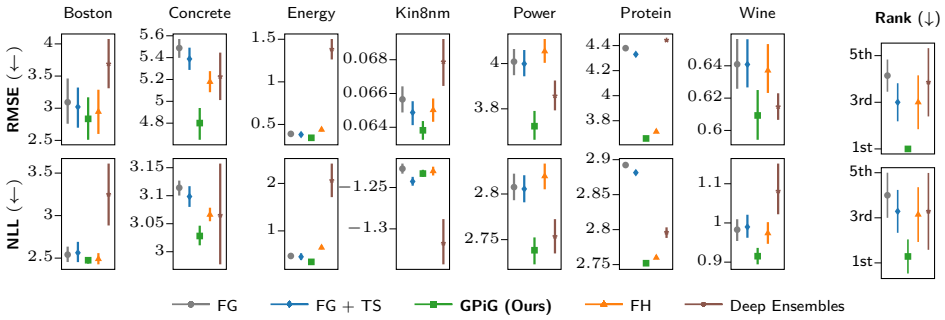

Figure 3: UCI regression benchmark results. Average ranks are computed across datasets.

distribution, $\Gamma^{-1}(1,1)$ (Chen et al., 2014); and our approach, (3) the GP induced Gaussian (GPiG) prior. We also compare BNNs against Deep Ensembles (Lakshminarayanan et al., 2017), a state-of-the-art approach for uncertainty estimation in deep learning (Ashukha et al., 2019; Ovadia et al., 2019). Furthermore, we additionally compare to the "cold" posterior (Wenzel et al., 2020) that uses a Fixed Gaussian prior and temperature scaling (FG+TS).

We use stochastic gradient Hamiltonian Monte Carlo (SGHMC) (Chen et al., 2014) for posterior inference with scale adaptation (Springenberg et al., 2016), where the hyperparameters of SGHMC are adjusted during a burn-in phase. We sample four independent chains; for each chain, the number of collected samples after thinning is 30 except for the Protein dataset for which we collect 60. We use a step size of 0.01 and a momentum coefficient of 0.01. For FH, we resample the prior variances using a Gibb step every 100 iterations.

Each dataset is randomly split into 90%/10% train/test sets. This splitting process is repeated 10 times except for the Protein dataset which uses 5 splits. We use a 2 hidden-layer BNN with tanh activation function, containing 100 hidden units for smaller datasets and 200 hidden units for the Protein dataset. We map a target hierarchical-GP prior with an RBF kernel to GPiG by using our proposed Wasserstein scheme. We place a prior $\text{LogNormal}(\log\sqrt{2D}, 1)$ on the length-scales and a prior $\text{LogNormal}(0.1, 1)$ on the variance. Here, D is the number of input dimensions.

Fig. 3 illustrates the average test NLL and RMSE. For most datasets, our optimized priors provide the best results and outperform Deep Ensembles in terms of both RMSE and NLL. We notice that tempering the posterior only shows small improvements on FG. Instead, by using our GPiG, the predictive performance of the true posterior improves considerably.

4. Concluding remarks and future work

In this paper, we analyzed the effect of placing a prior on the output of a BNN rather than on its parameters. To this goal, we borrowed the flexibility of the Wasserstein distance as similarity metric between functions sampled from GPs priors to functions sampled from BNNs. We tested this contribution on a classic regression benchmark and we showed sensible improvements over classic choices and state-of-the-art inference methods. On this premises, we expect to improve considerably also for deeper models, like CNN, and on tasks where having good uncertainty estimate is key for obtaining good results, such as active learning and Bayesian optimization.

References

Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *International Conference on Learning Representations*, 2019.

François-Xavier Briol, Chris J. Oates, Mark Girolami, Michael A. Osborne, and Dino Sejdinovic. Probabilistic Integration: A Role in Statistical Computation? *Statistical Science*, 34(1):1–22, 02 2019.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic Gradient Hamiltonian Monte Carlo. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 1683–1691, Bejing, China, 22–24 Jun 2014. PMLR.

Jon Cockayne, Chris J. Oates, Ilse C.F. Ipsen, and Mark Girolami. A Bayesian Conjugate Gradient Method (with Discussion). *Bayesian Analysis*, 14(3):937–1012, 09 2019.

Alexander G. de G. Matthews, Jiri Hron, Mark Rowland, Richard E. Turner, and Zoubin Ghahramani. Gaussian Process Behaviour in Wide Deep Neural Networks. In *International Conference on Learning Representations*, 2018.

Sylvain Delattre and Nicolas Fournier. On the Kozachenko–Leonenko entropy estimator. *Journal of Statistical Planning and Inference*, 185:69–93, 2017.

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

Daniel Flam-Shepherd, James Requeima, and David Duvenaud. Mapping Gaussian Process Priors to Bayesian Neural Networks. In *2nd NeurIPS workshop on Bayesian Deep Learning*, 2017.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR.

Zoubin Ghahramani and Carl E. Rasmussen. Bayesian Monte Carlo. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 505–512. MIT Press, 2003.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

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

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

Leonid Vitaliyevich Kantorovich. On the transfer of masses. *Doklady Akademii Nauk SSSR*, 37:227–229, 1942.

Leonid Vitaliyevich Kantorovich. On a problem of Monge. *Uspekhi Matematicheskikh Nauk*, 3:225–226, 1948.

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

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

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

Qiang Liu and Dilin Wang. Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2378–2386. Curran Associates, Inc., 2016.

Radford M Neal. Bayesian learning for neural networks. 1996.

A. O'Hagan. Bayes–Hermite quadrature. *Journal of Statistical Planning and Inference*, 29 (3):245 – 260, 1991.

Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz E Khan, Anirudh Jain, Runa Eschenhagen, Richard E Turner, and Rio Yokota. Practical Deep Learning with Bayesian Principles. In *Advances in Neural Information Processing Systems 32*, pages 4287–4299. Curran Associates, Inc., 2019.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pages 13991–14002, 2019.

Tim Pearce, Russell Tsuchida, Mohamed Zaki, Alexandra Brintrup, and Andy Neely. Expressive priors in Bayesian neural networks: Kernel combinations and periodic functions. In *35th Conference on Uncertainty in Artificial Intelligence*, 2019.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

Simone Rossi, Pietro Michiardi, and Maurizio Filippone. Good Initializations of Variational Bayes for Deep Models. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5487–5497, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Simone Rossi, Sebastien Marmin, and Maurizio Filippone. Walsh-Hadamard Variational Inference for Bayesian Deep Learning. In *Advances in Neural Information Processing Systems 33*. 2020.

Jost Tobias Springenberg, Aaron Klein, Stefan Falkner, and Frank Hutter. Bayesian optimization with robust bayesian neural networks. In *Advances in Neural Information Processing Systems*, pages 4134–4142, 2016.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1015–1022, Haifa, Israel, June 2010. Omnipress.

Florian Wenzel, Kevin Roth, Bastiaan S. Veeling, Jakub Świątkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How Good is the Bayes Posterior in Deep Neural Networks Really? In *37th International Conference on Machine Learning*, 2020.