Tractable Function-Space Variational Inference in Bayesian Neural Networks

Tim G. J. Rudner[*] University of Oxford
Zonghao Chen University College London
Yee Whye Teh University of Oxford
Yarin Gal University of Oxford

Abstract

Reliable predictive uncertainty estimation plays an important role in enabling the deployment of neural networks to safety-critical settings. A popular approach for estimating the predictive uncertainty of neural networks is to define a prior distribution over the network parameters, infer an approximate posterior distribution, and use it to make stochastic predictions. However, explicit inference over neural network parameters makes it difficult to incorporate meaningful prior information about the data generating process into the model. In this paper, we pursue an alternative approach. Recognizing that the primary object of interest in most settings is the distribution over functions induced by the posterior distribution over neural network parameters, we frame Bayesian inference in neural networks explicitly as inferring a posterior distribution over functions and propose a scalable function-space variational inference method that allows incorporating prior information and results in reliable predictive uncertainty estimates. We show that the proposed method leads to state-of-the-art uncertainty estimation and predictive performance on a range of prediction tasks and demonstrate that it performs well on a challenging safety-critical medical diagnosis task in which reliable uncertainty estimation is essential.

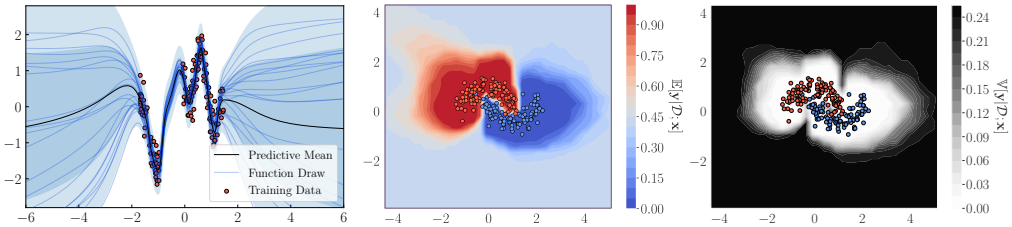
1 Introduction

Machine learning models succeed at an increasingly wide range of narrowly defined tasks [Krizhevsky et al., 2012, Mnih et al., 2013, Silver et al., 2016, Jumper et al., 2021], but fail without warning when used on inputs that are meaningfully different from the data they were trained on [Amodei et al., 2016, Hendrycks et al., 2021]. To deploy machine learning models in safety-critical environments where failures are costly or may endanger human lives, machine learning methods must be reliable and have the ability to 'fail gracefully.' As a promising tool for incorporating fail-safe mechanisms into machine learning systems, predictive uncertainty quantification allows machine learning models to express their confidence in the correctness of their predictions [Gal and Ghahramani, 2016].

In this paper, we develop a method for obtaining reliable uncertainty estimates in Bayesian neural networks (BNNs, Neal [1996]). While BNNs have promised to combine the advantages of deep learning and Bayesian inference, existing approaches for approximate inference in BNNs fall short of this promise and have been demonstrated to result in approximate posterior predictive distributions that underperform non-Bayesian methods both in terms of predictive accuracy and uncertainty quantification—making them of limited use in practice [Ovadia et al., 2019, Foong et al., 2019, Farquhar et al., 2020a, Band et al., 2021]. A potential reason for this shortcoming is that commonly used parameter-space inference methods make it difficult to define meaningful priors that effectively incorporate information about the data-generating process into inference.

[*]Corresponding author. Email: <tim.rudner@cs.ox.ac.uk>.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).



(a) Predictive Distribution **(b)** Predictive Mean **(c)** Predictive Variance

Figure 1: 1D regression on the *Snelson* dataset and binary classification on the *Two Moons* dataset. The plots show the predictive distributions of a BNNs, obtained via function-space variational inference (FSVI). For further illustrative exampled and comparisons to deep ensembles and BNNs learned via parameter-space variational inference, see Appendix B.

To avoid this limitation, we follow the foundation work by Sun et al. [2019] and consider a variational objective defined explicitly in terms of distributions over *functions* induced by distributions over parameters. In contrast to prior works that rely on approximation techniques that prevent such function-space variational objectives to be used with high-dimensional inputs and highly-overparameterized neural networks [Sun et al., 2019], we propose a simple estimator of the Kullback-Leibler divergence between distributions over functions that allows for stochastic variational inference. The proposed estimation procedure allows defining priors that explicitly encourage high predictive uncertainty away from the training data as well as priors that reflect relevant information about the task at hand.

We demonstrate that this approach leads to posterior approximations that exhibit significantly improved predictive uncertainty estimates compared to a wide array of state-of-the-art Bayesian and non-Bayesian methods.[2] Figure 1 shows examples of predictive distributions obtained via function-space variational inference on low-dimensional, easy-to-visualize datasets. As can be seen in the figures, the predictive distributions fit the training data well while also exhibiting a high degree of predictive uncertainty in parts of the input space far away from the training data, as desired.

Contributions. We propose a simple estimation procedure for performing function-space variational inference in BNNs. The variational method allows for the incorporation of meaningful prior information about the data-generating process into the inference and produces reliable predictive uncertainty estimates. We perform a thorough empirical evaluation in which we compare the proposed approach to a wide array of competing methods and show that it consistently results in high predictive performance and reliable predictive uncertainty estimates, outperforming other methods in terms of predictive accuracy, robustness to distribution shifts, and uncertainty-based detection of distributionally-shifted data samples. We evaluate the proposed method on standard benchmarking datasets as well as on a safety-critical medical diagnosis task in which reliable uncertainty estimation is essential.

2 Preliminaries

We consider supervised learning tasks on data $\mathcal{D} \doteq \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N = (\mathbf{X}_\mathcal{D}, \mathbf{y}_\mathcal{D})$ with inputs $\mathbf{x}_n \in \mathcal{X} \subseteq \mathbb{R}^D$ and targets $\mathbf{y}_n \in \mathcal{Y}$, where $\mathcal{Y} \subseteq \mathbb{R}^Q$ for regression and $\mathcal{Y} \subseteq \{0, 1\}^Q$ for classification tasks. Bayesian neural networks (BNNs) are stochastic neural networks trained using (approximate) Bayesian inference. Denoting the parameters of such a stochastic neural network by the multivariate random variable $\mathbf{\Theta} \in \mathbb{R}^P$ and letting the function mapping defined by a neural network architecture be given by $f : \mathcal{X} \times \mathbb{R}^P \to \mathbb{R}^Q$, we obtain a random function $f(\cdot\,; \mathbf{\Theta})$. For a parameter realization $\boldsymbol{\theta}$, we obtain a corresponding function realization, $f(\cdot\,; \boldsymbol{\theta})$. When evaluated at a finite collection of points $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^m$, $f(\mathbf{X}; \mathbf{\Theta})$ is a multivariate random variable and $f(\mathbf{X}; \boldsymbol{\theta})$ is a vector.

Letting $p(\mathbf{y}_\mathcal{D} \mid f(\mathbf{X}_\mathcal{D}; \boldsymbol{\theta}))$ be the likelihood of observing the targets $\mathbf{y}$ under the stochastic function $f(\cdot\,; \mathbf{\Theta})$ evaluated at inputs $\mathbf{X}_\mathcal{D}$ and letting $p_\mathbf{\Theta}$ be a prior distribution over the stochastic network parameters $\mathbf{\Theta}$, we can use Bayesian inference for finding the conditional distribution over parameters given the observed data, $p_{\mathbf{\Theta}|\mathcal{D}}$ [MacKay, 1992, Neal, 1996]. However, since the mapping f is a non-

[2] Our code can be accessed at https://github.com/timrudner/FSVI.

linear function of the stochastic parameters $\boldsymbol{\Theta}$, exact inference is analytically intractable. Variational inference is an approach that seeks to sidestep this intractability by framing posterior inference as a variational optimization problem, where the goal is to find a distribution $q_{\boldsymbol{\Theta}}$ in a variational family $\mathcal{Q}_{q_{\boldsymbol{\Theta}}}$ that solves the variational problem $\min_{q_{\boldsymbol{\Theta}} \in \mathcal{Q}_{q_{\boldsymbol{\Theta}}}} \mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\Theta}} \parallel p_{\boldsymbol{\Theta}|\mathcal{D}})$ [Wainwright and Jordan, 2008]. If $\mathcal{Q}_{q_{\boldsymbol{\Theta}}}$ is the family of mean-field Gaussian distributions and the prior distribution over parameters $p_{\boldsymbol{\Theta}}$ given by a diagonal Gaussian distribution, the resulting variational objective is amenable to stochastic variational inference and can be optimized using gradient-based methods. [Hinton and van Camp, 1993, Graves, 2011, Hoffman et al., 2013, Blundell et al., 2015].

2.1 A Function-Space Perspective on Variational Inference in Bayesian Neural Networks

By framing variational inference in BNNs as computing an approximation to a posterior distribution over *functions*, $p_{f(\cdot\,;\boldsymbol{\Theta})|\mathcal{D}}$, we can incorporate meaningful prior information about the underlying data-generating process into training. Specifically, given a prior distribution over functions $p_{f(\cdot;\boldsymbol{\Theta})}$, induced by a prior distribution over parameters, $p_{\boldsymbol{\Theta}}$, we can express the problem of finding a posterior distribution over functions, $p_{f(\cdot;\boldsymbol{\Theta})|\mathcal{D}}$, variationally as

$$\min_{q_{\boldsymbol{\Theta}} \in \mathcal{Q}_{q_{\boldsymbol{\Theta}}}} \mathbb{D}_{\mathrm{KL}}(q_{f(\cdot\,;\boldsymbol{\Theta})} \,\|\, p_{f(\cdot\,;\boldsymbol{\Theta})|\mathcal{D}}),$$

where $q_{f(\cdot\,;\boldsymbol{\Theta})}$ is a variational distribution over functions induced by a variational distribution over parameters $q_{\boldsymbol{\Theta}}$ and $p_{f(\cdot\,;\boldsymbol{\Theta})|\mathcal{D}}$ is the posterior distribution over functions induced by the posterior distribution over parameters $p_{\boldsymbol{\Theta})|\mathcal{D}}$. As discussed by Burt et al. [2021], this variational objective is guaranteed to be well-defined for suitably chosen prior distributions over functions. Specifically, the KL divergence between two distributions over functions generated from different distributions over parameters applied to the same mapping (e.g., the same neural network architecture) is well-defined (i.e., finite) if the KL divergence between the distributions over parameters is finite, since, by the strong data processing inequality [Polyanskiy and Wu, 2017],

$$\mathbb{D}_{\mathrm{KL}}(q_{f(\cdot\,;\boldsymbol{\Theta})} \,\|\, p_{f(\cdot;\boldsymbol{\Theta})}) \leq \mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\Theta}} \,\|\, p_{\boldsymbol{\Theta}}). \tag{1}$$

As a result, if $\mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\Theta}} \,\|\, p_{\boldsymbol{\Theta}}) < \infty$, which is the case for finite-dimensional parameter vectors $\boldsymbol{\Theta}$ and $q_{\boldsymbol{\Theta}}$ absolutely continuous with respect to $p_{\boldsymbol{\Theta}}$, then the function-space KL divergence is finite and thus well-defined as a variational objective.

Hence, for a likelihood function defined on a finite set of training targets $\mathbf{y}_{\mathcal{D}}$ and a suitably defined prior distribution over functions, we can express the variational problem above equivalently as the well-defined maximization problem $\max_{q_{\boldsymbol{\Theta}} \in \mathcal{Q}_{\boldsymbol{\theta}}} \mathcal{F}(q_{\boldsymbol{\Theta}}, \mathbf{X}_{\mathcal{D}}, \mathbf{y}_{\mathcal{D}})$ with

$$\mathcal{F}(q_{\boldsymbol{\Theta}}, \mathcal{D}) \doteq \mathbb{E}_{q_{f(\mathbf{X}_{\mathcal{D}};\boldsymbol{\Theta})}}[\log p(\mathbf{y}_{\mathcal{D}} \mid f(\mathbf{X}_{\mathcal{D}}; \boldsymbol{\theta}))] - \mathbb{D}_{\mathrm{KL}}(q_{f(\cdot;\boldsymbol{\Theta})} \,\|\, p_{f(\cdot;\boldsymbol{\Theta})}), \tag{2}$$

where $\mathbb{D}_{\mathrm{KL}}(q_{f(\cdot;\boldsymbol{\Theta})} \,\|\, p_{f(\cdot;\boldsymbol{\Theta})})$ is also a KL divergence between distributions over functions.

Unfortunately, evaluating the KL divergence in Equation (2) is intractable in general. To obtain a tractable objective, Sun et al. [2019] show that $\mathbb{D}_{\mathrm{KL}}(q_{f(\cdot;\boldsymbol{\Theta})} \,\|\, p_{f(\cdot;\boldsymbol{\Theta})})$ can be expressed as the supremum of the KL divergence from $q_{f(\cdot;\boldsymbol{\Theta})}$ to $p_{f(\cdot;\boldsymbol{\Theta})}$ over all *finite* sets of evaluation points, resulting in the objective function

$$\mathcal{F}(q_{\boldsymbol{\Theta}}, \mathcal{D}) = \mathbb{E}_{q_{f(\mathbf{X}_{\mathcal{D}};\boldsymbol{\Theta})}}[\log p(\mathbf{y}_{\mathcal{D}} \mid f(\mathbf{X}_{\mathcal{D}}; \boldsymbol{\theta}))] - \sup_{\mathbf{X} \in \mathcal{X}_{\mathbb{N}}} \mathbb{D}_{\mathrm{KL}}(q_{f(\mathbf{X};\boldsymbol{\Theta})} \,\|\, p_{f(\mathbf{X};\boldsymbol{\Theta})}), \tag{3}$$

where $\mathcal{X}_{\mathbb{N}} \doteq \bigcup_{n \in \mathbb{N}} \{\mathbf{X} \in \mathcal{X}_n \mid \mathcal{X}_n \subseteq \mathbb{R}^{n \times D}\}$ is the collection of all finite sets of evaluation points. However, this objective function is still challenging to optimize in practice: The supremum cannot be obtained analytically and—more problematically—the KL divergence term itself is analytically intractable and difficult to estimate—even for a single evaluation point. While the supremum can be estimated via Monte Carlo sampling, the fact that the KL divergence between BNNs cannot be easily estimated has prevented the application of function-space variational inference to challenging prediction tasks. For example, Sun et al. [2019] attempted to circumvent estimating the KL divergence between BNNs directly by instead approximating the gradients of the KL divergence with respect to the variational parameters via a spectral Stein gradient estimation. However, this approach does not scale to large neural networks or high-dimensional input data.

In the next section, we will describe an approximation procedure that allows scaling function-space variational inference to large networks and high-dimensional input data.

3 A Tractable Function-Space Variational Objective

The primary obstacle to making the objective in Equation (2) tractable is the KL divergence from $q_{f(\cdot;\boldsymbol{\Theta})}$ to $p_{f(\cdot;\boldsymbol{\Theta})}$. There are two reasons why the KL divergence in Equation (3) is intractable: (**i**) For BNNs or other non-linear models, we do not have access to the probability density functions of the multivariate distributions $q_{f(\mathbf{X};\boldsymbol{\Theta})}$ and $p_{f(\mathbf{X};\boldsymbol{\Theta})}$; (**ii**) For all but extremely simple input spaces, we are unable to compute the supremum over all possible finite sets of evaluation points. In the remainder of this section, we outline a simple approach for obtaining an estimator of the KL divergence that allows for scalable gradient-based optimization of Equation (3).

We first approach the problem of computing the KL divergence between two BNNs evaluated at a finite set of points. To do so, we first derive tractable approximations to the distributions over functions $q_{f(\mathbf{X};\boldsymbol{\Theta})}$ and $p_{f(\mathbf{X};\boldsymbol{\Theta})}$ Next, we show that under these approximations, we are able to obtain a closed-form expression of the KL divergence and propose a simple Monte Carlo estimator to approximate the supremum in the function-space KL divergence.

3.1 Approximating Distributions over Functions via Local Linearization

To obtain an approximation to the probability distributions of $q_{f(\mathbf{X};\boldsymbol{\Theta})}$ and $p_{f(\mathbf{X};\boldsymbol{\Theta})}$, we use a first-order Taylor expansion of the *mapping* f about the mean parameters of $q_{\boldsymbol{\Theta}}$ and $p_{\boldsymbol{\Theta}}$, respectively, and derive the induced distributions under the linearized mapping.

For a stochastic function $f(\cdot\,;\boldsymbol{\Theta})$ defined in terms of stochastic parameters $\boldsymbol{\Theta}$ distributed according to distribution $g_{\boldsymbol{\Theta}}$ with $\mathbf{m} \doteq \mathbb{E}_{g_{\boldsymbol{\Theta}}}[\boldsymbol{\Theta}]$ and $\mathbf{S} \doteq \mathrm{Cov}_{g_{\boldsymbol{\Theta}}}[\boldsymbol{\Theta}]$, we denote the linearization of the stochastic function $f(\cdot\,;\boldsymbol{\Theta})$ about $\mathbf{m}$ by

$$f(\cdot\,;\boldsymbol{\Theta}) \approx \tilde{f}(\cdot\,;\mathbf{m},\boldsymbol{\Theta}) \doteq f(\cdot\,;\mathbf{m}) + \mathcal{J}(\cdot,\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m}), \tag{4}$$

where $\mathcal{J}(\cdot,\mathbf{m}) \doteq (\partial f(\cdot\,;\boldsymbol{\Theta})/\partial\boldsymbol{\Theta})|_{\boldsymbol{\Theta}=\mathbf{m}}$ is the Jacobian of $f(\cdot\,;\boldsymbol{\Theta})$ evaluated at $\boldsymbol{\Theta}=\mathbf{m}$, and the mean and covariance of the distribution over the linearized mapping $\tilde{f}$ at $\mathbf{X},\mathbf{X}' \in \mathcal{X}$ are given by

$$\mathbb{E}[\tilde{f}(\mathbf{X};\boldsymbol{\Theta})] = f(\mathbf{X};\mathbf{m}) \tag{5}$$

$$\mathrm{Cov}[\tilde{f}(\mathbf{X};\boldsymbol{\Theta}),\tilde{f}(\mathbf{X}';\boldsymbol{\Theta})] = \mathcal{J}(\mathbf{X},\mathbf{m})\mathbf{S}\mathcal{J}(\mathbf{X}',\mathbf{m})^\top. \tag{6}$$

For a derivation of this result, see Appendix A. Since Gaussianity is preserved under affine transformations, if $g_{\boldsymbol{\Theta}}$ is a multivariate Gaussian distribution with mean $\mathbf{m}$ and diagonal co-variance $\mathbf{S}$, then the distribution $\tilde{g}$ over $\tilde{f}(\mathbf{X}\,;\boldsymbol{\Theta})$ is given by

$$\tilde{g}_{\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\Theta})} = \mathcal{N}(f(\mathbf{X};\mathbf{m}),\mathcal{J}(\mathbf{X},\mathbf{m})\mathbf{S}\mathcal{J}(\mathbf{X}',\mathbf{m})^\top). \tag{7}$$

For stochastic functions parameterized by many millions of parameters, obtaining the covariance of $\tilde{g}_{\tilde{f}(\mathbf{X};\boldsymbol{\Theta})}$—which requires computing an inner product of two Jacobian matrices—can be computationally expensive. Instead of computing the distribution over the linearized mapping exactly, we can compute a Monte Carlo estimate. To construct a suitable Monte Carlo estimator, we consider a partition of the set of parameters into sets α and β (with $|\beta| \ll |\alpha|$) and note that the linearized mapping can then be expressed as

$$\tilde{f}(\cdot\,;\mathbf{m},\boldsymbol{\Theta}) = f(\cdot\,;\mathbf{m}) + \tilde{f}_\alpha(\cdot\,;\mathbf{m},\boldsymbol{\Theta}_\alpha) + \mathcal{J}(\cdot,\mathbf{m})_\beta(\boldsymbol{\Theta}_\beta - \mathbf{m}_\beta), \tag{8}$$

with

$$\tilde{f}_\alpha(\cdot\,;\mathbf{m},\boldsymbol{\Theta}_\alpha) \doteq \mathcal{J}(\cdot,\mathbf{m})_\alpha(\boldsymbol{\Theta}_\alpha - \mathbf{m}_\alpha), \tag{9}$$

where $\mathcal{J}(\cdot,\mathbf{m})_\alpha$ and $\mathcal{J}(\cdot,\mathbf{m})_\beta$ are the columns of the Jacobian matrix corresponding to the sets of parameters α and β, respectively, and $\boldsymbol{\Theta}_\alpha$ and $\boldsymbol{\Theta}_\beta$ are the corresponding random parameter vectors. Noting that Equation (8) expresses $\tilde{f}$ as a sum of (affine transformations of) random variables, we can use the fact that for independent Gaussian random variables $\mathbf{X}$ and $\mathbf{Y}$, the distribution $h_{\mathbf{Z}}$ of $\mathbf{Z}=\mathbf{X}+\mathbf{Y}$ is equal to the convolution of the distributions $h_{\mathbf{X}}$ and $h_{\mathbf{Y}}$ to obtain an approximation to $\tilde{f}$. In particular, we can show that if $g_{\boldsymbol{\Theta}}$ is a multivariate Gaussian distribution with $\boldsymbol{\Theta}_\alpha \perp \boldsymbol{\Theta}_\beta$, the distribution $\tilde{g}_{\tilde{f}(\mathbf{X};\boldsymbol{\Theta})}$ can be approximated by the Monte Carlo estimator

$$\hat{\tilde{g}}_{\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\Theta})} = \mathcal{N}\Big(f(\mathbf{X};\mathbf{m}) + \frac{1}{R}\sum_{j=1}^{R}\tilde{f}_\alpha(\mathbf{X};\mathbf{m},\boldsymbol{\Theta}_\alpha)^{(j)},\mathcal{J}(\mathbf{X},\mathbf{m})_\beta\mathbf{S}_\beta\mathcal{J}(\mathbf{X}',\mathbf{m})_\beta{}^\top\Big), \tag{10}$$

where $g_{\boldsymbol{\Theta}_\beta} = \mathcal{N}(\mathbf{m}_\beta,\mathbf{S}_\beta)$ and samples $\tilde{f}_\alpha(\mathbf{X};\mathbf{m},\boldsymbol{\Theta}_\alpha)^{(j)}$ are obtained by sampling parameters from the distribution $g_{\boldsymbol{\Theta}_\alpha} = \mathcal{N}(\mathbf{m}_\alpha,\mathbf{S}_\alpha)$. For a derivation of this result, see Appendix A. In our empirical evaluation, we chose α to be the set of parameters in neural network layers $1:L-1$ and β to be the set of parameters in the final neural network layer.

3.2 Approximating the Function-Space Kullback-Leibler Divergence

From Section 3.1, we know that if $q_{\boldsymbol{\Theta}}$ and $p_{\boldsymbol{\Theta}}$ are both Gaussian distributions, then the induced distributions under the linearized mapping $\tilde{f}$ evaluated at a finite set of evaluation points will be Gaussian as well. This means that for Gaussian variational and prior distributions over $\boldsymbol{\Theta}$, we can obtain locally accurate approximations to the induced distributions $q_{f(\cdot;\boldsymbol{\Theta})}$ to $p_{f(\cdot;\boldsymbol{\Theta})}$ and use them to approximate the KL divergence in the variational objective by $\mathbb{D}_{\mathrm{KL}}(\tilde{q}_{\tilde{f}(\mathbf{X};\boldsymbol{\Theta})} \,\|\, \tilde{p}_{\tilde{f}(\mathbf{X};\boldsymbol{\Theta})})$. Moreover, for an isotropic Gaussian prior and a mean-field Gaussian variational distribution, $\mathbb{D}_{\mathrm{KL}}(\tilde{q}_{\tilde{f}(\mathbf{X};\boldsymbol{\Theta})} \,\|\, \tilde{p}_{\tilde{f}(\mathbf{X};\boldsymbol{\Theta})})$ is a KL divergence between two multivariate Gaussians can be obtained analytically.

Using this approximation, we obtain an estimator of the variational objective given by

$$\tilde{\mathcal{F}}(q_{\boldsymbol{\Theta}}, \mathcal{D}) \doteq \mathbb{E}_{q_{f(\mathbf{X}_{\mathcal{D}};\boldsymbol{\Theta})}}[\log p(\mathbf{y}_{\mathcal{D}} \mid f(\mathbf{X}_{\mathcal{D}}; \boldsymbol{\theta}))] - \sup_{\mathbf{X} \in \mathcal{X}_{\mathbb{N}}} \mathbb{D}_{\mathrm{KL}}(\tilde{q}_{\tilde{f}(\mathbf{X};\boldsymbol{\Theta})} \,\|\, \tilde{p}_{\tilde{f}(\mathbf{X};\boldsymbol{\Theta})}), \tag{11}$$

where the arguments of the KL divergence have been replaced by the (locally accurate) approximations to the variational and prior distributions over functions evaluated at $\mathbf{X}$, respectively. Since the stochastic functions $\tilde{f}(\cdot\,; \boldsymbol{\Theta})$ induced by $q_{\boldsymbol{\Theta}}$ and $p_{\boldsymbol{\Theta}}$ under the linearized mapping will be closer to the stochastic function under f the smaller the variance of $q_{\boldsymbol{\Theta}}$ and $p_{\boldsymbol{\Theta}}$, respectively, the approximation to the KL divergence will be more accurate the smaller the variance of $q_{\boldsymbol{\Theta}}$ and $p_{\boldsymbol{\Theta}}$.

Next, we turn to computing the supremum. Unlike Sun et al. [2019], who consider the supremum as a separate optimization problem, we do not seek to compute the supremum by searching over points $\mathbf{X} \in \mathcal{X}_{\mathbb{N}}$ but instead propose to estimate the supremum at every gradient step via a simple finite-sample estimator. Specifically, letting $I(\mathbf{X}) \doteq \mathbb{D}_{\mathrm{KL}}(\tilde{q}_{\tilde{f}(\mathbf{X};\boldsymbol{\Theta})} \,\|\, \tilde{p}_{\tilde{f}(\mathbf{X};\boldsymbol{\Theta})})$, we estimate $G = \sup_{\mathbf{X} \in \mathcal{X}_{\mathbb{N}}} I(\mathbf{X})$ using the Monte Carlo estimator

$$\hat{G}(\mathcal{X}_{\mathcal{C}}^S) = \max_{\mathbf{X} \in \mathcal{X}_{\mathcal{C}}^S} I(\mathbf{X}), \tag{12}$$

where $\mathcal{X}_{\mathcal{C}}^S \doteq \{\mathbf{X}_{\mathcal{C}}^{(i)}\}_{i=1}^S$ is a collection of S sets of *context points* $\mathbf{X}_{\mathcal{C}}^{(i)} \doteq \{\mathbf{x}^{(j)}\}_{j=1}^K$ jointly sampled from a context distribution $p_{\mathcal{X}_{\mathcal{C}}}$. Each context set $\mathbf{X}_{\mathcal{C}}^{(i)}$ can be viewed as a single Monte Carlo sample from the input space so that the estimator $\hat{G}(\mathcal{X}_{\mathcal{C}}^S)$ provides an S-sample Monte Carlo estimate of the supremum. While this estimator is crude and only provides a rough approximation to the true supremum, it encourages the variational distribution over functions to match the prior distribution over functions on the sets of context points. The choice of the context distribution $p_{\mathcal{X}_{\mathcal{C}}}$ can be informed by knowledge about the prediction task and should be viewed as a problem-specific modeling choice. Similarly, the numbers of samples S and K are hyperparameters to be optimized with a validation set. For details on how $p_{\mathcal{X}_{\mathcal{C}}}$ is chosen in the empirical evaluation in Section 5, see Appendix E.

3.3 Stochastic Estimation of the Function-Space Variational Objective

Let $q_{\boldsymbol{\Theta}}$ be a Gaussian mean-field variational distribution, let $p_{\boldsymbol{\Theta}}$ be an isotropic Gaussian prior, let $(\mathbf{X}_{\mathcal{B}}, \mathbf{y}_{\mathcal{B}})$ be a mini-batch of the training data, and reparameterize $\boldsymbol{\Theta}$ as $\hat{\boldsymbol{\Theta}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\epsilon}^{(j)}) \doteq \boldsymbol{\mu} + \boldsymbol{\Sigma} \odot \boldsymbol{\epsilon}^{(j)}$. Using the estimator $\hat{G}(\mathcal{X}_{\mathcal{C}}^S)$ defined above and estimating the expected log-likelihood via Monte Carlo sampling, we obtain a Monte Carlo estimator for the function-space variational objective:

$$\bar{\mathcal{F}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathcal{X}_{\mathcal{C}}^S, \mathcal{D}) = \frac{1}{M} \sum_{j=1}^{M} \log p(\mathbf{y}_{\mathcal{B}} \mid f(\mathbf{X}_{\mathcal{B}}; \hat{\boldsymbol{\Theta}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\epsilon}^{(j)}))) - \max_{\mathbf{X} \in \mathcal{X}_{\mathcal{C}}^S} \mathbb{D}_{\mathrm{KL}}(\tilde{q}_{\tilde{f}(\mathbf{X};\hat{\boldsymbol{\Theta}})} \,\|\, \tilde{p}_{\tilde{f}(\mathbf{X};\hat{\boldsymbol{\Theta}})})$$

with $\boldsymbol{\epsilon}^{(j)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_P)$ and $\mathcal{X}_{\mathcal{C}}^S$ as defined above. Crucially, to avoid having to perform expensive matrix operations as part of the KL computation, we make a diagonalization approximation to the covariance matrices of $\tilde{q}$ and $\tilde{p}$, respectively.

Selection of Prior. For all experiments that involve uncertainty quantification, we chose a prior distribution over parameters that induces a prior distribution over functions $p_{f(\cdot;\boldsymbol{\Theta})}$ that, under the likelihood, exhibits a high degree of predictive uncertainty. Such a prior will encourage the BNN to have high uncertainty at evaluation points from regions in input space where $p_{\mathcal{X}_{\mathcal{C}}}$ has non-zero support and, under smoothness constraints, on evaluation points in nearby regions. For settings where prior information is encoded in data—for example, in the form of expert demonstrations of robotic manipulation tasks, or in the form of pre-trained networks in continual or transfer learning—an empirical prior that reflect this information can be specified. For further details, see Appendix E, and for sensitivity studies on the effect of different prior choices on performance, see Appendix D.

Selection of Context Distribution. The distribution $p_{\mathcal{X}_c}$ allows us to incorporate information about the data-generating process into training. This way, $p_{\mathcal{X}_c}$ allows us obtain gradient estimates that encourage the variational distribution to match the prior over functions in more relevant parts of the input space. By taking advantage of the abundance of data available in real-world settings, we can construct context distributions from large datasets like ImageNet [Krizhevsky et al., 2012] or from small but diverse datasets like CIFAR-100. In our experiments, we choose two types of context distributions. One distribution constructed from the training data and only containing randomly sampled monochrome images, and one constructed from a related real-world dataset generated from a similar but different data distribution. For further details, see Appendix E, and for sensitivity studies on the effect of different context distributions on performance, see Appendix D.

Posterior Predictive Distribution. After optimizing the variational objective with respect to the parameters of the variational distribution $q_{\boldsymbol{\Theta}}$, we obtain an approximate posterior predictive distribution

$$q(\mathbf{y}_* \mid \mathbf{x}_*) = \int p(\mathbf{y}_* \mid f(\mathbf{x}_*; \boldsymbol{\theta}))\, q_{f(\mathbf{x}_*;\boldsymbol{\Theta})}\, \mathrm{d}f(\mathbf{x}_*; \boldsymbol{\theta}) \approx \frac{1}{M_*} \sum_{j=1}^{M_*} p(\mathbf{y}_* \mid f(\mathbf{x}_*; \boldsymbol{\Theta}^{(j)})), \boldsymbol{\Theta}^{(j)} \sim q_{\boldsymbol{\Theta}}, \tag{13}$$

where M_* is the number of Monte Carlo samples used to estimate the predictive distribution.

Computational Complexity. Under the diagonal approximation used to compute $\bar{\mathcal{F}}$, the time and space complexity for stochastic gradient estimation scale linearly in the mini-batch size $n_{\mathcal{B}}$ and the number forward passes through the network per input point (i.e., the number of Monte Carlo samples M used to estimate the expected log-likelihood plus the number of points on which the KL divergence is evaluated, SK), yielding a time and space complexity of $\mathcal{O}(n_{\mathcal{B}} p(M + SK))$. The time complexity of a prediction on n_* points with M_* Monte Carlo samples has a complexity of $\mathcal{O}(n_* M_*)$.

4 Related Work

There is a growing body of work on function-space approaches to inference in BNNs, deep learning, and applications such as continual learning [Benjamin et al., 2019, Sun et al., 2019, Titsias et al., 2020, Burt et al., 2021, Pan et al., 2020, Ma and Hernández-Lobato, 2021, Rudner et al., 2022].

Function-Space Inference in Bayesian Neural Networks. Previously proposed methods for FSVI in BNNs are based on approximate gradient estimators and either replace the supremum in Equation (3) with an expectation [Sun et al., 2019] or do not define an explicit variational objective [Wang et al., 2019]. Furthermore, [Sun et al., 2019] rely on pretrained Gaussian process priors over functions for which the function-space variational inference problem is not well-defined (see Section 2.1 and Burt et al. [2021]). More recent work has attempted to circumvent the intractability of the variational objective in Equation (2) by proposing alternative objectives for function-space inference in BNNs [Ma et al., 2019, Carvalho et al., 2020, Ober and Aitchison, 2020, Ma and Hernández-Lobato, 2021]. Rudner et al. [2022] extend the approach presented in Section 3 to sequential inference problems and apply it to continual learning.

Linear Models. Immer et al. [2020] and Khan et al. [2019] show that approximate BNN posterior distribution via the Laplace and Generalized-Gauss-Newton approximation corresponds to exact posteriors under linearizations of different models. Unlike in our approach, they use a Laplace approximation and do not perform variational inference and do not optimize the variance parameters. Furthermore, Immer et al. [2020] and Khan et al. [2019] use a neural network model to obtain a parameter maximum a posteriori estimate, but then use a linearization of the neural network model to compute a posterior predictive distribution. In contrast, our work only uses the linearization to obtain an estimator of the variational objective but uses the unlinearized model to construct a posterior predictive distribution.

Pathologies of Variational Inference in Bayesian Neural Networks. Burt et al. [2021] consider the function-space variational objective in Equation (2) and show that the KL divergence between BNNs with different networks architectures are not well-defined. A parallel line of research showed that posterior predictive distributions of shallow BNNs with mean-field variational distributions have a limited ability to represent complex covariance structures in function space [Foong et al., 2019, 2020] but that deep BNNs do not suffer from this limitation [Farquhar et al., 2020b]. Our results are consistent with the findings of Farquhar et al. [2020b] that mean-field variational distributions are able to represent complex covariance structures in function space.

Table 1: Comparison of in- and out-of-distribution performance metrics on FashionMNIST (mean ± standard error over ten random seeds). The last two columns show the AUROC for binary in- vs. out-of-distribution detection on MNIST (M) and NotMNIST (NM). MNIST and NotMNIST are used as out-of-distribution datasets. Best overall results for single and ensemble models are printed in boldface with gray shading. Results within a 95% confidence interval of the best overall result are printed in boldface only. All methods use the same four-layer CNN architecture. For further details about model architectures and training and evaluation protocols, see Appendix E.

Method	Accuracy ↑	NLL ↓	ECE ↓	AUROC M ↑	AUROC NM ↑
MAP	91.73±0.08	0.288±0.003	0.037±0.001	87.00±0.30	74.85±1.31
MFVI [Blundell et al., 2015]	91.03±0.04	0.354±0.003	0.038±0.001	93.10±0.34	88.88±0.74
MFVI (tempered)	91.38±0.05	0.519±0.005	0.058±0.001	86.30±0.29	80.78±0.68
MFVI (radial) [Farquhar et al., 2020a]	90.31±0.11	0.340±0.001	0.035±0.001	84.40±0.68	82.11±1.15
MC DROPOUT	90.55±0.04	0.230±0.001	**0.012**±0.001	88.46±0.57	80.02±1.04
SWAG [Maddox et al., 2019]	92.56±0.05	0.300±0.000	0.043±0.001	85.18±0.35	80.31±0.30
DUQ [van Amersfoort et al., 2020]	92.40±0.20	–	–	95.50±0.70	94.60±1.80
BNN-LAPLACE [Immer et al., 2020]	92.25±0.10	0.244±0.003	**0.012**±0.003	95.55±0.60	–
SPG [Ma and Hernández-Lobato, 2021]	91.60±0.14	0.254±0.050	–	95.60±6.00	–
FSVI ($p_{\mathbf{X}_C}$ = random monochrome)	**93.15**±0.13	0.203±0.004	**0.014**±0.002	96.55±0.41	95.15±0.71
FSVI ($p_{\mathbf{X}_C}$ = KMNIST)	**93.15**±0.15	**0.180**±0.005	**0.012**±0.002	**99.81**±0.19	**98.39**±0.21
Deep Ensemble	92.49±0.01	0.242±0.001	**0.019**±0.000	89.22±0.09	83.17±0.91
FSVI Ensemble	**94.44**±0.07	**0.181**±0.001	0.020±0.001	**97.85**±0.15	**96.95**±0.20

5 Empirical Evaluation

The seminal work of Sun et al. [2019] introduced function-space variational inference as a tool for approximate inference in BNNs but the approach proposed in Sun et al. [2019] is limited to small neural networks and low-dimensional input data [Ma and Hernández-Lobato, 2021]. In this section, we evaluate FSVI on high-dimensional classification tasks that were previously out of reach under the approach proposed by Sun et al. [2019] and compare FSVI to several well-established and state-of-the-art methods. We show that FSVI (sometimes *significantly*) outperforms existing Bayesian and non-Bayesian methods in terms of their in-distribution uncertainty calibration and out-of-distribution predictive uncertainty estimation. For a details on models, training and validation procedures, and datasets used, see Appendix E. For a comparison to Sun et al. [2019] on small-scale regression problems, see Appendix B.2.

5.1 Predictive Performance, Uncertainty Estimation, and Distribution Shift Detection

In this set of experiments, we assess the reliability of the uncertainty estimates generated by FSVI. If a BNN trained via FSVI is able to perform reliable uncertainty estimation, its predictive uncertainty will be significantly higher on input points that were generated according to a different data-generating distribution than the training data. For models trained on the FashionMNIST dataset, we use the MNIST and NotMNIST datasets as out-of-distribution evaluation points, while for models trained on the CIFAR-10 dataset, we use the SVHN dataset as out-of-distribution evaluation points.

For models trained on either FashionMNIST or CIFAR-10, we evaluate their in-distribution performance in terms of test accuracy, test log-likelihood, and test calibration. To evaluate the quality of different models' uncertainty estimates, we compute uncertainty estimates for the pairs FashionMNIST/MNIST, FashionMNIST/NotMNIST, and CIFAR-10/SVHN to and measure for a range of thresholds how well the datasets in each pair can be separated solely based on the uncertainty estimates. This experiment setup follows prior work by van Amersfoort et al. [2020] and Immer et al. [2020]. We report the area under the receiver operating characteristic (ROC) curve in Tables 1 and 2.

Predictive Performance and Calibration. To assess in-distribution predictive performance and calibration, we report the test accuracy, negative log-likelihood (NLL), and expected calibration error (ECE) for models trained on FashionMNIST and CIFAR-10 in Tables 1 and 2. On both FashionMNIST and CIFAR-10, FSVI achieves the lowest NLL and either the best or second-best predictive accuracy and ECE, respectively, across all methods. Notably, FSVI significantly outperforms SPG [Ma and Hernández-Lobato, 2021], an alternative function-space variational inference method.

Predictive Uncertainty under Distribution Shift. In Tables 1 and 2, we report evaluation metrics that elucidate the reliability of different methods' predictive uncertainty under distribution shift.

Table 2: Comparison of in- and out-of-distribution performance metrics on CIFAR-10 (mean ± standard error over ten random seeds). SVHN and corrupted CIFAR-10 (C-CIFAR) are used as an out-of-distribution datasets. The penultimate column shows the AUROC for binary in- vs. out-of-distribution detection on SVHN. Best overall results for single and ensemble models are printed in boldface with gray shading. Results within a 95% confidence interval of the best overall result are printed in boldface only. All methods use a ResNet-18 architecture. For further details about model architectures and training and evaluation protocols, see Appendix E.

Method	Accuracy↑	NLL↓	ECE↓	OOD-AUROC↑	C-CIFAR Acc↑
MAP	92.19±0.15	0.307±0.006	0.046±0.001	95.17±0.40	78.55±1.01
MFVI [Blundell et al., 2015]	89.98±0.09	0.340±0.006	0.040±0.001	92.14±0.34	79.36±1.35
MFVI (tempered)	90.87±0.11	0.360±0.003	0.048±0.001	91.82±0.90	79.86±1.32
MC DROPOUT	91.32±0.06	0.310±0.004	0.041±0.001	90.32±0.57	80.19±1.44
SWAG [Maddox et al., 2019]	93.13±0.14	0.480±0.001	0.067±0.002	89.79±0.50	76.12±0.51
VOGN [Osawa et al., 2019]	84.27±0.20	0.477±0.006	0.040±0.002	87.60±0.20	–
DUQ [van Amersfoort et al., 2020]	**94.10**±0.20	–	–	92.70±1.30	–
SPG [Ma and Hernández-Lobato, 2021]	77.69±0.64	0.675±0.030	–	88.30±4.00	–
FSVI ($p_{\mathbf{X}_C}$ = random monochrome)	92.21±0.04	**0.299**±0.003	0.035±0.001	94.57±0.24	**80.76**±0.48
FSVI ($p_{\mathbf{X}_C}$ = CIFAR-100)	92.27±0.04	**0.287**±0.003	**0.028**±0.001	**98.02**±0.10	**81.21**±0.47
Deep Ensemble	95.13±0.06	0.158±0.001	0.019±0.001	**98.04**±0.07	**81.22**±0.37
FSVI Ensemble	**95.19**±0.03	**0.156**±0.003	**0.013**±0.001	**99.19**±0.41	**81.35**±0.48

FSVI exhibits reliable predictive uncertainty estimates that allow distinguishing between in- and out-of-distribution inputs with high accuracy. As would be expected, we observe that using context distributions that reflect our knowledge about the data-generating process can significantly improve uncertainty quantification under FSVI. For the FashionMNIST experiment, we used the KMNIST dataset, which contains grayscale images of Kuzushiji letters, and for the CIFAR-10 experiment, we used the CIFAR-100 dataset, which contains RGB images of 100 classes. Both KMNIST and CIFAR-100 differ from the OOD datasets (MNIST and NotMNIST and SVHN, respectively) used to compute OOD-AUROC metrics in Tables 1 and 2, but using them as context distributions significantly increased the ability of BNNs trained via FSVI to identify distributionally shifted samples. Since the variational objective encourages matching the prior (which we chose to have high variance) on samples from the context distribution can improve uncertainty estimation in regions of the input space far from the training data.

5.2 Generalization and Reliability of Predictive Uncertainty under Distribution Shift

To assess the reliability of predictive models in deep learning, Ovadia et al. [2019] propose the following desiderata: In order for a model to be considered reliable, it ought to (i) exhibit low predictive uncertainty on training data and high predictive uncertainty on out-of-distribution inputs, (ii) generate predictive uncertainty estimates that allow distinguishing in- from out-of-distribution inputs, and (iii) if possible, maintain high predictive accuracy even under distribution shift. Models that satisfy these desiderata are less likely to make poor, high-confidence predictions and more amenable for use in safety-critical downstream tasks.

To illustrate these desiderata, we follow Ovadia et al. [2019] and consider the rotated MNIST task, where a model is trained on MNIST and evaluated on rotated MNIST digits. The goal is to maintain a high level of predictive accuracy (measured in terms of Brier scores) while exhibiting an increasing level of predictive uncertainty on distribution shifts of increasing magnitude. Figure 2 shows Brier scores (lower is better) and predictive entropy estimates (higher means more uncertain) of four different models. As rotating the MNIST digits gradually shifts the data distributions, we would expect Brier scores to increase (corresponding to worse predictive accuracy) as the rotation angle increases. A model with reliable predictive entropy estimates would only experience a small decrease under distribution shift while exhibiting a large increase in predictive uncertainty. As can be seen in the plot, the Brier scores of FSVI

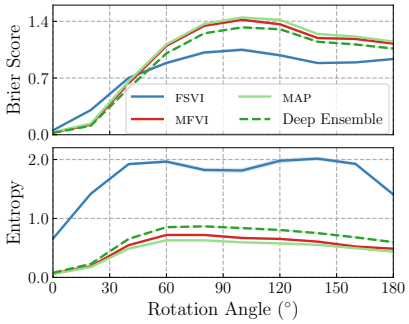
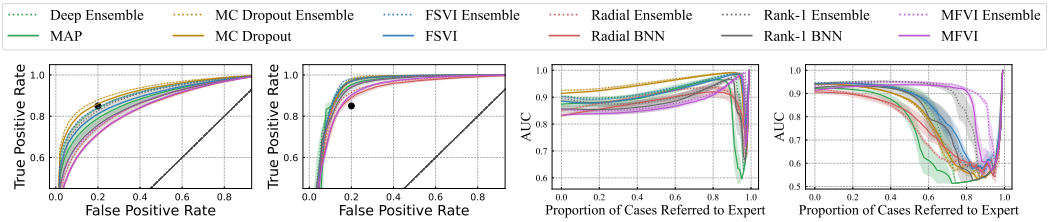


Figure 2: Predictive uncertainty and accuracy on rotated MNIST. Models with reliable uncertainty estimates would exhibit higher predictive uncertainty the more the digits are rotated. Ideally, such models would maintain high predictive accuracy (low Brier score).



(a) ROC: In-Domain **(b)** ROC: Country Shift **(c)** Selective Prediction AUROC: In-Domain **(d)** Selective Prediction AUROC: Country Shift

Figure 4: We jointly assess model predictive performance and uncertainty quantification on both in-domain and distributionally shifted data. **Left:** The *receiver operating characteristic curve* (ROC) for in-population diagnosis on the (**a**) EyePACS [2015] test set and for (**b**) changing medical equipment and patient populations on the APTOS [2019] test set. The dot in **black** denotes the NHS-recommended 85% sensitivity and 80% specificity ratios [Widdowson, 2016]. **Right:** *Selective prediction* on AUROC in (**c**) EyePACS [2015] and (**d**) APTOS [2019] settings. Shading denotes standard error over six random seeds. See Appendix B.1 for tabular results.

decreases the least, while FSVI's uncertainty is significantly higher than other models'. To assess the reliability of different uncertainty quantification methods on a more challenging distribution-shift task, we consider corrupted CIFAR-10 inputs under the second-mildest corruption level used in [Ovadia et al., 2019] and report our results in Table 2. Consistent with the rotated MNIST results, FSVI achieves the highest accuracy on the corrupted data.

5.3 Safety-Critical Uncertainty-Aware Selective Prediction: Diabetic Retinopathy Diagnosis

To evaluate the reliability of the predictive uncertainty of FSVI in a real-world safety-critical setting, we consider the task of diagnosing diabetic retinopathy (DR), a medical condition that can lead to impaired vision, from retina scans [Leibig et al., 2017, Filos et al., 2019, Band et al., 2021]. We use two publicly available datasets, EyePACS [2015] and APTOS [2019], each containing RGB images of a human retina graded by a medical expert on the following scale: 0 (no DR), 1 (mild DR), 2 (moderate DR), 3 (severe DR), and 4 (proliferative DR). The Kaggle dataset was collected from patients in the United States, while the APTOS dataset was collected from patients in India using cheaper but more modern scanning devices. We follow Leibig et al. [2017], Filos et al. [2019], and Band et al. [2021] and binarize all examples from both the EyePACS and APTOS datasets by dividing the classes up into sight-threatening diabetic retinopathy—defined as moderate diabetic retinopathy or worse (classes $\{2, 3, 4\}$)—and non-sight-threatening diabetic retinopathy—defined as no or mild diabetic retinopathy (classes $\{0, 1\}$). This results in a binary prediction task.

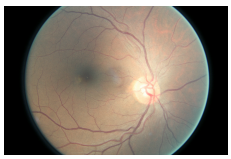


Figure 3: Retina scan examples. **Top:** healthy. **Bottom:** unhealthy.

To assess the reliability of predictive models when medical training and test data are obtained from different patient populations or collected with the same medical equipment, we follow Band et al. [2021] and use the Kaggle dataset for training and the distributionally shifted APTOS dataset for evaluation. The results are shown in Figure 4, which plot the ROC curves for the binary prediction problems as well as the area under the ROC curve for an uncertainty aware selective prediction task. For further details about the uncertainty-aware selective prediction evaluation protocol, see Appendix E.4. Figure 4 shows that FSVI performs well on all four tasks and is only outperformed by MC DROPOUT. For full tabular results, see Appendix B.1.

6 Conclusion

The paper proposed a scalable and effective approach to function-space variational inference in BNNs. We demonstrated that the proposed estimator of the function-space variational objective can be scaled up to high-dimensional data and large neural network architectures and that FSVI exhibits consistently reliable in- and out-of-distribution predictive performance on a wide range of datasets when compared to well-established and state-of-the-art uncertainty quantification methods. We hope that this work will lead to further research into function-space variational inference and the development of more sophisticated data-driven prior distributions over functions.

Acknowledgements

We thank Bryn Elesedy, Bobby He, and Andrew Jesson for feedback on an early draft of this paper, and we thank Joost van Amersfoort for helpful discussions on experiment design and implementations. Tim G. J. Rudner is funded by the Rhodes Trust and the Engineering and Physical Sciences Research Council (EPSRC). We gratefully acknowledge donations of computing resources by the Alan Turing Institute.

References

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25:*, pages 1106–1114, 2012.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In *NIPS Deep Learning Workshop*. 2013.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. 529, 2016.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Zidek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596 (7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety, 2016.

Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved problems in ml safety, 2021.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML 2016, pages 1050–1059, 2016.

Radford M Neal. Bayesian Learning for Neural Networks. 1996.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. 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 32*. 2019.

Andrew Y. K. Foong, Yingzhen Li, José Miguel Hernández-Lobato, and Richard E. Turner. 'in-between' uncertainty in Bayesian neural networks, 2019.

Sebastian Farquhar, Michael A. Osborne, and Yarin Gal. Radial Bayesian neural networks: Beyond discrete support in large-scale Bayesian deep learning. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1352–1362. PMLR, 26–28 Aug 2020a.

Neil Band, Tim G. J. Rudner, Qixuan Feng, Angelos Filos, Zachary Nado, Michael W. Dusenberry, Ghassen Jerfel, Dustin Tran, and Yarin Gal. Benchmarking Bayesian Deep Learning on Diabetic Retinopathy Detection Tasks. 2021.

Shengyang Sun, Guodong Zhang, Jiaxin Shi, and Roger B. Grosse. Functional variational Bayesian neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

David J. C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Comput.*, 4(3):448–472, May 1992. ISSN 0899-7667. doi: 10.1162/neco.1992.4.3.448.

Martin J Wainwright and Michael I Jordan. *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc., Hanover, MA, USA, 2008. ISBN 1601981848.

Geoffrey E. Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, COLT '93, page 5–13, New York, NY, USA, 1993. Association for Computing Machinery. ISBN 0897916115.

Alex Graves. Practical variational inference for neural networks. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, page 2348–2356, Red Hook, NY, USA, 2011. Curran Associates Inc. ISBN 9781618395993.

Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, May 2013. ISSN 1532-4435.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. volume 37 of *Proceedings of Machine Learning Research*, pages 1613–1622, Lille, France, 07–09 Jul 2015. PMLR.

David R. Burt, Sebastian W. Ober, Adrià Garriga-Alonso, and Mark van der Wilk. Understanding variational inference in function-space. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2021.

Yury Polyanskiy and Yihong Wu. Strong data-processing inequalities for channels and Bayesian networks. In Eric Carlen, Mokshay Madiman, and Elisabeth M. Werner, editors, *Convexity and Concentration*, pages 211–249, New York, NY, 2017. Springer New York. ISBN 978-1-4939-7005-6.

Ari Benjamin, David Rolnick, and Konrad Kording. Measuring and regularizing networks in function space. In *International Conference on Learning Representations*, 2019.

Michalis K. Titsias, Jonathan Schwarz, Alexander G. de G. Matthews, Razvan Pascanu, and Yee Whye Teh. Functional regularisation for continual learning with Gaussian processes. In *International Conference on Learning Representations*, 2020.

Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard E. Turner, and Mohammad Emtiyaz Khan. Continual deep learning by functional regularisation of memorable past. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Chao Ma and José Miguel Hernández-Lobato. Functional variational inference based on stochastic process generators. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

Tim G. J. Rudner, Freddie Bickford Smith, Qixuan Feng, Yee Whye Teh, and Yarin Gal. Continual Learning via Sequential Function-Space Variational Inference. In *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2022.

Ziyu Wang, Tongzheng Ren, Jun Zhu, and Bo Zhang. Function space particle optimization for Bayesian neural networks. In *International Conference on Learning Representations*, 2019.

Chao Ma, Yingzhen Li, and Jose Miguel Hernandez-Lobato. Variational implicit processes. 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 4222–4233. PMLR, 09–15 Jun 2019.

Eduardo D. C. Carvalho, Ronald Clark, Andrea Nicastro, and Paul H. J. Kelly. Scalable uncertainty for computer vision with functional variational inference. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

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

Alexander Immer, Maciej Korzepa, and Matthias Bauer. Improving predictions of Bayesian neural networks via local linearization, 2020.

Mohammad Emtiyaz E Khan, Alexander Immer, Ehsan Abedi, and Maciej Korzepa. Approximate inference turns deep networks into Gaussian processes. In *Advances in Neural Information Processing Systems 32*, pages 3094–3104. Curran Associates, Inc., 2019.

Andrew Y. K. Foong, David R. Burt, Yingzhen Li, and Richard E. Turner. On the expressiveness of approximate inference in Bayesian neural networks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Sebastian Farquhar, Lewis Smith, and Yarin Gal. Liberty or depth: Deep Bayesian neural nets do not need complex weight posterior approximations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020b.

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

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for Bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, pages 13153–13164, 2019.

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*, volume 32, pages 4287–4299. Curran Associates, Inc., 2019.

Christian Leibig, Vaneeda Allken, Murat Seçkin Ayhan, Philipp Berens, and Siegfried Wahl. Leveraging Uncertainty Information From Deep Neural Networks for Disease Detection. *Nature Scientific Reports*, 7(1):17816, 2017.

Angelos Filos, Sebastian Farquhar, Aidan N. Gomez, Tim G. J. Rudner, Zachary Kenton, Lewis Smith, Milad Alizadeh, Arnoud de Kroon, and Yarin Gal. A systematic comparison of Bayesian deep learning robustness in diabetic retinopathy tasks, 2019.

EyePACS. Diabetic Retinopathy Detection Dataset, 2015.

APTOS. APTOS 2019 Blindness Detection Dataset, 2019.

D. T. S. Widdowson. The Management of Grading Quality: Good Practice in the Quality Assurance of Grading. *Tech. Rep.*, 2016.

Alexander G. de G. Matthews, James Hensman, Richard Turner, and Zoubin Ghahramani. On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. volume 51 of *Proceedings of Machine Learning Research*, pages 231–239, Cadiz, Spain, 09–11 May 2016. PMLR.

M. J. Schervish. *Theory of Statistics*. Springer-Verlag, New York, NY, 1995.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1257–1264. MIT Press, 2006.

Joost van Amersfoort, Lewis Smith, Andrew Jesson, Oscar Key, and Yarin Gal. Variational deterministic uncertainty quantification, 2021.

Pavel Izmailov, Wesley J. Maddox, Polina Kirichenko, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Subspace inference for Bayesian deep learning. In Ryan P. Adams and Vibhav Gogate, editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 1169–1179. PMLR, 22–25 Jul 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 5.
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Appendix A.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix A.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix E.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Table 1, Table 2, and Table 3.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix E.9.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See `https://github.com/timrudner/FSVI`.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Supplementary Material

Table of Contents

A Proofs & Derivations **15**

A.1 Function-Space Variational Objective 15

A.2 Distribution under Linearized Function Mapping 16

B Further Empirical Results **19**

B.1 Tabular Results for Diabetic Retinopathy Diagnosis Tasks 19

B.2 UCI Regression 19

C Illustrative Examples **20**

C.1 Two Moons Classification Problem 20

C.2 Synthetic 1D Regression Problems 21

D Sensitivity Studies **22**

D.1 Ablation Study on the Effect of Different Function-Space Priors 22

D.2 Context Distribution 23

D.3 Prior Variance 24

D.4 Number of Sets of Context Points 25

D.5 Number of Context Points 26

E Implementation, Training, and Evaluation Details **27**

E.1 Hyperparameter Selection Protocol 27

E.2 FashionMNIST vs. MNIST/NotMNIST 27

E.3 CIFAR-10 vs. SVHN 27

E.4 Diabetic Retinopathy Diagnosis 27

E.5 Two Moons 28

E.6 1D Regression 28

E.7 Further Implementation Details 28

E.8 Selection of Context Distribution 28

E.9 Compute Resources 28

Appendix A Proofs & Derivations

A.1 Function-Space Variational Objective

This proof follows steps from Matthews et al. [2016]. Consider measures $\hat{P}$ and P both of which define distributions over some function f, indexed by an infinite index set X. Let $\mathcal{D}$ be a dataset and let $\mathbf{X}_\mathcal{D}$ denote a set of inputs and $\mathbf{y}_\mathcal{D}$ a set of targets. Consider the measure-theoretic version of Bayes' Theorem [Schervish, 1995]:

$$\frac{d\hat{P}}{dP}(f) = \frac{p_X(Y \mid f)}{p(Y)}, \tag{A.1}$$

where $p_X(Y \mid f)$ is the likelihood and $p(Y) = \int_{\mathbb{R}^X} p_X(Y \mid f) dP(f)$ is the marginal likelihood. We assume that the likelihood function is evaluated at a finite subset of the index set X. Denote by $\pi_C : \mathbb{R}^X \to \mathbb{R}^C$ a projection function that takes a function and returns the same function, evaluated at a finite set of points C, so we can write

$$\frac{d\hat{P}}{dP}(f) = \frac{d\hat{P}_{\mathbf{X}_\mathcal{D}}}{dP_{\mathbf{X}_\mathcal{D}}}(\pi_{\mathbf{X}_\mathcal{D}}(f)) = \frac{p(\mathbf{y}_\mathcal{D} \mid \pi_{\mathbf{X}_\mathcal{D}}(f))}{p(\mathbf{y}_\mathcal{D})}, \tag{A.2}$$

and similarly, the marginal likelihood becomes $p(\mathbf{y}_\mathcal{D}) = \int p(\mathbf{y}_\mathcal{D} \mid f_{\mathbf{X}_\mathcal{D}}) dP_{\mathbf{X}_\mathcal{D}}(f_{\mathbf{X}_\mathcal{D}})$. Now, considering the measure-theoretic version of the KL divergence between an approximating stochastic process Q and a posterior stochastic process $\hat{P}$, we can write

$$\mathbb{D}_{\text{KL}}(Q \,\|\, \hat{P}) = \int \log \frac{dQ}{dP}(f) dQ(f) - \int \log \frac{d\hat{P}}{dP}(f) dQ(f), \tag{A.3}$$

where P is some prior stochastic process. Now, we can apply the measure-theoretic Bayes' Theorem to obtain

$$\begin{aligned}
\mathbb{D}_{\text{KL}}(Q \,\|\, \hat{P}) &= \int \log \frac{dQ}{dP}(f) dQ(f) - \int \log \frac{d\hat{P}}{dP}(f) dQ(f) && \text{(A.4)}\\
&= \int \log \frac{dQ^\pi}{dP^\pi}(f) dQ^\pi(f) - \int \log \frac{d\hat{P}_{\mathbf{X}_\mathcal{D}}}{dP_{\mathbf{X}_\mathcal{D}}}(f_{\mathbf{X}_\mathcal{D}}) dQ_{\mathbf{X}_\mathcal{D}}(f_{\mathbf{X}_\mathcal{D}}) && \text{(A.5)}\\
&= \int \log \frac{dQ^\pi}{dP^\pi}(f) dQ^\pi(f) - \mathbb{E}_{Q_{\mathbf{X}_\mathcal{D}}}\left[\log p\left(\mathbf{y}_\mathcal{D} \mid f_{\mathbf{X}_\mathcal{D}}\right)\right] - \log p(\mathbf{y}_\mathcal{D}), && \text{(A.6)}
\end{aligned}$$

where $\frac{dQ^\pi}{dP^\pi}(f)$ is marginally consistent given the projection π. Rearranging, we can get

$$\begin{aligned}
p(\mathbf{y}_\mathcal{D}) &= \mathbb{E}_{Q_{\mathbf{X}_\mathcal{D}}}\left[\log p\left(\mathbf{y}_\mathcal{D} \mid f_{\mathbf{X}_\mathcal{D}}\right)\right] - \int \log \frac{dQ^\pi}{dP^\pi}(f) dQ^\pi(f) + \mathbb{D}_{\text{KL}}(Q^\pi \,\|\, \hat{P}) && \text{(A.7)}\\
&\geq \mathbb{E}_{Q_{\mathbf{X}_\mathcal{D}}}\left[\log p\left(\mathbf{y}_\mathcal{D} \mid f_{\mathbf{X}_\mathcal{D}}\right)\right] - \int \log \frac{dQ^\pi}{dP^\pi}(f) dQ^\pi(f) && \text{(A.8)}\\
&= \mathbb{E}_{Q_{\mathbf{X}_\mathcal{D}}}\left[\log p\left(\mathbf{y}_\mathcal{D} \mid f_{\mathbf{X}_\mathcal{D}}\right)\right] - \mathbb{D}_{\text{KL}}(Q^\pi \,\|\, P^\pi). && \text{(A.9)}
\end{aligned}$$

Slightly abusing notation (ignoring the fact that that there is no infinite-dimensional Lebesgue measure), we can express this lower bound as

$$\begin{aligned}
p(\mathbf{y}_\mathcal{D}) &\geq \mathbb{E}_{Q_{\mathbf{X}_\mathcal{D}}}\left[\log p\left(\mathbf{y}_\mathcal{D} \mid f_{\mathbf{X}_\mathcal{D}}\right)\right] - \int \log \frac{dQ^\pi}{dP^\pi}(f) dQ^\pi(f) && \text{(A.10)}\\
&= \mathbb{E}_{Q_{\mathbf{X}_\mathcal{D}}}\left[\log p\left(\mathbf{y}_\mathcal{D} \mid f_{\mathbf{X}_\mathcal{D}}\right)\right] - \mathbb{D}_{\text{KL}}(Q_{\mathbf{X}_\mathcal{D}, \mathbf{X}_{\setminus\mathcal{D}}} \,\|\, P_{\mathbf{X}_\mathcal{D}, \mathbf{X}_{\setminus\mathcal{D}}}), && \text{(A.11)}
\end{aligned}$$

where $\mathbf{X}_{\setminus\mathcal{D}}$ is an infinite index set excluding the finite index set $\mathbf{X}_\mathcal{D}$, that is, $\mathbf{X}_{\setminus\mathcal{D}} \cap \mathbf{X}_\mathcal{D} = \varnothing$. By Theorem 1 in Sun et al. [2019] and the chain rule of KL divergence, we then obtain the lower bound

$$p(\mathbf{y}_\mathcal{D}) \geq \mathbb{E}_{Q_{\mathbf{X}_\mathcal{D}}}\left[\log p\left(\mathbf{y}_\mathcal{D} \mid f_{\mathbf{X}_\mathcal{D}}\right)\right] - \sup_{\mathbf{X} \in \mathcal{X}_\mathbb{N}} \mathbb{D}_{\text{KL}}(Q_\mathbf{X} \,\|\, P_\mathbf{X}), \tag{A.12}$$

where $\mathcal{X}_\mathbb{N} \doteq \bigcup_{n \in \mathbb{N}} \{\mathbf{X} \in \mathcal{X}_n \mid \mathcal{X}_n \subseteq \mathbb{R}^{n \times D}\}$ is the collection of all finite sets of evaluation points.

A.2 Distribution under Linearized Function Mapping

Proposition 1 (Distribution under Linearized Mapping)**.** *For a stochastic function $f(\cdot\,;\boldsymbol{\Theta})$ defined in terms of stochastic parameters $\boldsymbol{\Theta}$ distributed according to distribution $g_{\boldsymbol{\Theta}}$ with $\mathbf{m} \doteq \mathbb{E}_{g_{\boldsymbol{\Theta}}}[\boldsymbol{\Theta}]$ and $\mathbf{S} \doteq Cov_{g_{\boldsymbol{\Theta}}}[\boldsymbol{\Theta}]$, denote the linearization of the stochastic function $f(\cdot\,;\boldsymbol{\Theta})$ about $\mathbf{m}$ by*

$$f(\cdot\,;\boldsymbol{\Theta}) \approx \tilde{f}(\cdot\,;\mathbf{m},\boldsymbol{\Theta}) \doteq f(\cdot\,;\mathbf{m}) + \mathcal{J}(\cdot,\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m}),$$

where $\mathcal{J}(\cdot,\mathbf{m}) \doteq (\partial f(\cdot\,;\boldsymbol{\Theta})/\partial\boldsymbol{\Theta})|_{\boldsymbol{\Theta}=\mathbf{m}}$ is the Jacobian of $f(\cdot\,;\boldsymbol{\Theta})$ evaluated at $\boldsymbol{\Theta}=\mathbf{m}$. Then the mean and co-variance of the distribution over the linearized mapping $\tilde{f}$ at $\mathbf{X},\mathbf{X}' \in \mathcal{X}$ are given by

$$\mathbb{E}[\tilde{f}(\mathbf{X};\boldsymbol{\Theta})] = f(\mathbf{X};\mathbf{m})$$
$$\mathrm{Cov}[\tilde{f}(\mathbf{X};\boldsymbol{\Theta}),\tilde{f}(\mathbf{X}';\boldsymbol{\Theta})] = \mathcal{J}(\mathbf{X},\mathbf{m})\mathbf{S}\mathcal{J}(\mathbf{X}',\mathbf{m})^\top.$$

Proof. We wish to find $\mathbb{E}[\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\Theta})]$ and

$$\mathrm{Cov}(\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\Theta}),\tilde{f}(\mathbf{X}';\mathbf{m},\boldsymbol{\theta})) = \mathbb{E}[(\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\theta}) - \mathbb{E}[\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\theta})])\,(\tilde{f}(\mathbf{X}';\mathbf{m},\boldsymbol{\theta}) - \mathbb{E}[\tilde{f}(\mathbf{X}';\mathbf{m},\boldsymbol{\theta})])^\top]. \tag{A.13}$$

To see that $\mathbb{E}[\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\theta})] = f(\mathbf{X};\mathbf{m})$, note that, by linearity of expectation, we have

$$\begin{aligned}\mathbb{E}[\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\theta})] &= \mathbb{E}[f(\mathbf{X};\mathbf{m}) + \mathcal{J}(\mathbf{X};\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m})] \\ &= f(\mathbf{X};\mathbf{m}) + \mathcal{J}(\mathbf{X};\mathbf{m})(\mathbb{E}[\boldsymbol{\Theta}]-\mathbf{m}) = f(\mathbf{X};\mathbf{m}).\end{aligned} \tag{A.14}$$

To see that $\mathrm{Cov}(\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\theta}),\tilde{f}(\mathbf{X}';\mathbf{m},\boldsymbol{\theta})) = \mathcal{J}(\mathbf{X};\mathbf{m})\mathbf{S}\mathcal{J}(\mathbf{X}';\mathbf{m})^\top$, note that in general, for a multivariate random variable $\mathbf{Z}$, $\mathrm{Cov}(\mathbf{Z},\mathbf{Z}) = \mathbb{E}[\mathbf{Z}\mathbf{Z}^\top] + \mathbb{E}[\mathbf{Z}]\mathbb{E}[\mathbf{Z}]^\top$, and hence,

$$\mathrm{Cov}(\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\Theta}),\tilde{f}(\mathbf{X}';\mathbf{m},\boldsymbol{\Theta})) = \mathbb{E}[\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\Theta})\tilde{f}(\mathbf{X}';\mathbf{m},\boldsymbol{\Theta})^\top] - \mathbb{E}[\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\Theta})]\mathbb{E}[\tilde{f}(\mathbf{X}';\mathbf{m},\boldsymbol{\Theta})]^\top. \tag{A.15}$$

We already know that $\mathbb{E}[\tilde{f}(\mathbf{X};\boldsymbol{\Theta})] = f(\mathbf{X};\mathbf{m})$, so we only need to find $\mathbb{E}[\tilde{f}(\mathbf{X};\boldsymbol{\Theta})\tilde{f}(\mathbf{X}';\boldsymbol{\Theta})^\top]$:

$$\begin{aligned}&\mathbb{E}_{g_{\boldsymbol{\Theta}}}[\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\Theta})\tilde{f}(\mathbf{X}';\mathbf{m},\boldsymbol{\Theta})^\top] \\ &=\mathbb{E}_{g_{\boldsymbol{\Theta}}}[(f(\mathbf{X};\mathbf{m}) + \mathcal{J}(\mathbf{X};\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m}))(f(\mathbf{X}';\mathbf{m}) + \mathcal{J}(\mathbf{X}';\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m}))^\top] \qquad \text{(A.16)}\\ &=\mathbb{E}_{g_{\boldsymbol{\Theta}}}[f(\mathbf{X};\mathbf{m})f(\mathbf{X}';\mathbf{m})^\top + (\mathcal{J}(\mathbf{X};\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m}))(\mathcal{J}(\mathbf{X}';\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m}))^\top \\ &\qquad + f(\mathbf{X};\mathbf{m})(\mathcal{J}(\mathbf{X}';\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m}))^\top + \mathcal{J}(\mathbf{X};\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m})f(\mathbf{X}';\mathbf{m})^\top] \qquad \text{(A.17)}\\ &=\mathbb{E}_{g_{\boldsymbol{\Theta}}}[f(\mathbf{X};\mathbf{m})f(\mathbf{X}';\mathbf{m})^\top + \mathcal{J}(\mathbf{X};\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m})^\top\mathcal{J}(\mathbf{X}';\mathbf{m})^\top \\ &\qquad + f(\mathbf{X};\mathbf{m})(\mathcal{J}(\mathbf{X}';\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m}))^\top + \mathcal{J}(\mathbf{X};\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m})f(\mathbf{X}';\mathbf{m})^\top] \qquad \text{(A.18)}\\ &=f(\mathbf{X};\mathbf{m})f(\mathbf{X}';\mathbf{m})^\top + \mathcal{J}(\mathbf{X};\mathbf{m})\mathbb{E}_{g_{\boldsymbol{\Theta}}}[(\boldsymbol{\Theta}-\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m})^\top]\mathcal{J}(\mathbf{X}';\mathbf{m})^\top \\ &\qquad + f(\mathbf{X};\mathbf{m})(\mathcal{J}(\mathbf{X}';\mathbf{m})(\underbrace{\mathbb{E}_{g_{\boldsymbol{\Theta}}}[\boldsymbol{\Theta}]-\mathbf{m}}_{=0}))^\top + \mathcal{J}(\mathbf{X};\mathbf{m})(\underbrace{\mathbb{E}_{g_{\boldsymbol{\Theta}}}[\boldsymbol{\Theta}]-\mathbf{m}}_{=0})f(\mathbf{X}';\mathbf{m})^\top, \qquad \text{(A.19)}\end{aligned}$$

where the last line follows from the definition of $g_{\boldsymbol{\Theta}}$. By definition of the variance, we then obtain

$$\begin{aligned}\mathbb{E}_{g_{\boldsymbol{\Theta}}}[\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\Theta})\tilde{f}(\mathbf{X}';\mathbf{m},\boldsymbol{\Theta})^\top] &= f(\mathbf{X};\mathbf{m})f(\mathbf{X}';\mathbf{m})^\top + \mathcal{J}(\mathbf{X};\mathbf{m})\mathbb{E}_{g_{\boldsymbol{\Theta}}}[(\boldsymbol{\Theta}-\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m})^\top]\mathcal{J}(\mathbf{X}';\mathbf{m})^\top \\ &= f(\mathbf{X};\mathbf{m})f(\mathbf{X};\mathbf{m})^\top + \mathcal{J}(\mathbf{X};\mathbf{m})\mathrm{Cov}(\boldsymbol{\Theta})\mathcal{J}(\mathbf{X}';\mathbf{m})^\top.\end{aligned} \tag{A.20}$$

With this result, we obtain the covariance function

$$\begin{aligned}&\mathrm{Cov}(\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\Theta}),\tilde{f}(\mathbf{X}';\mathbf{m},\boldsymbol{\Theta})) \\ &= \mathbb{E}[\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\Theta})\tilde{f}(\mathbf{X}';\mathbf{m},\boldsymbol{\Theta})^\top] - \mathbb{E}[\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\Theta})]\mathbb{E}[\tilde{f}(\mathbf{X}';\mathbf{m},\boldsymbol{\Theta})]^\top \qquad \text{(A.21)}\\ &= \mathbb{E}[\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\Theta})\tilde{f}(\mathbf{X}';\mathbf{m},\boldsymbol{\Theta})^\top] - f(\mathbf{X};\mathbf{m})f(\mathbf{X};\mathbf{m})^\top + \mathcal{J}(\mathbf{X};\mathbf{m})\mathrm{Cov}(\boldsymbol{\Theta})\mathcal{J}(\mathbf{X}';\mathbf{m})^\top \qquad \text{(A.22)}\\ &= f(\mathbf{X};\boldsymbol{\Theta})f(\mathbf{X}';\boldsymbol{\Theta})^\top - f(\mathbf{X};\mathbf{m})f(\mathbf{X};\mathbf{m})^\top + \mathcal{J}(\mathbf{X};\mathbf{m})\mathrm{Cov}\boldsymbol{\Theta})\mathcal{J}(\mathbf{X}';\mathbf{m})^\top \qquad \text{(A.23)}\\ &= \mathcal{J}(\mathbf{X};\mathbf{m})\,\mathbb{V}[\boldsymbol{\Theta}]\mathcal{J}(\mathbf{X}';\mathbf{m})^\top. \qquad \text{(A.24)}\end{aligned}$$

Finally, $\mathrm{Cov}(\boldsymbol{\Theta}) = \mathbf{S}$ yields $\mathrm{Cov}(\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\Theta}),\tilde{f}(\mathbf{X}';\mathbf{m},\boldsymbol{\Theta})) = \mathcal{J}(\mathbf{X};\mathbf{m})\mathbf{S}\mathcal{J}(\mathbf{X}';\mathbf{m})^\top$. This concludes the proof. □

Proposition 2 (Approximate Distribution under Linearized Mapping)**.** *For a stochastic function $f(\cdot\,;\boldsymbol{\Theta})$ defined in terms of stochastic parameters $\boldsymbol{\Theta}$ distributed according to distribution $g_{\boldsymbol{\Theta}} = \mathcal{N}(\mathbf{m}, \mathbf{S})$, denote the linearization of the stochastic function $f(\cdot\,;\boldsymbol{\Theta})$ about $\mathbf{m}$ by*

$$f(\cdot\,;\boldsymbol{\Theta}) \approx \tilde{f}(\cdot\,;\mathbf{m},\boldsymbol{\Theta}) \doteq f(\cdot\,;\mathbf{m}) + \mathcal{J}(\cdot,\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m}),$$

where $\mathcal{J}(\cdot,\mathbf{m}) \doteq (\partial f(\cdot\,;\boldsymbol{\Theta})/\partial\boldsymbol{\Theta})|_{\boldsymbol{\Theta}=\mathbf{m}}$ is the Jacobian of $f(\cdot\,;\boldsymbol{\Theta})$ evaluated at $\boldsymbol{\Theta}=\mathbf{m}$. Then, for a partition of the set of parameters into sets α and β, a distribution $g_{\boldsymbol{\Theta}} = \mathcal{N}(\mathbf{m},\mathbf{S})$ with $\boldsymbol{\Theta}_\alpha \perp \boldsymbol{\Theta}_\beta$, the distribution $\tilde{g}_{\tilde{f}(\mathbf{X};\boldsymbol{\Theta})}$ can be approximated via the Monte Carlo estimator

$$\hat{\tilde{g}}_{\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\Theta})} = \mathcal{N}\Big(f(\mathbf{X};\mathbf{m}) + \frac{1}{R}\sum_{j=1}^{R} \tilde{f}_\alpha(\mathbf{X};\mathbf{m},\boldsymbol{\Theta}_\alpha)^{(j)}, \mathcal{J}(\mathbf{X},\mathbf{m})_\beta \mathbf{S}_\beta \mathcal{J}(\mathbf{X}',\mathbf{m})_\beta{}^\top\Big), \tag{A.25}$$

where $g_{\boldsymbol{\Theta}_\alpha} = \mathcal{N}(\mathbf{m}_\alpha,\mathbf{S}_\alpha)$, $g_{\boldsymbol{\Theta}_\beta} = \mathcal{N}(\mathbf{m}_\beta,\mathbf{S}_\beta)$, and

$$\tilde{f}_\alpha(\cdot\,;\mathbf{m},\boldsymbol{\Theta}_\alpha) \doteq \mathcal{J}(\cdot,\mathbf{m})_\alpha(\boldsymbol{\Theta}_\alpha - \mathbf{m}_\alpha), \tag{A.26}$$

with $\mathcal{J}(\cdot,\mathbf{m})_\alpha$ denoting the columns of the Jacobian matrix corresponding to the sets of parameters α and $\tilde{f}_\alpha(\mathbf{X};\mathbf{m},\boldsymbol{\Theta}_\alpha)^{(j)}$ for $j = 1,...,R$ obtained by sampling parameters from the distribution $g_{\boldsymbol{\Theta}_\alpha} = \mathcal{N}(\mathbf{m}_\alpha,\mathbf{S}_\alpha)$.

Proof. Consider a partition of the set of parameters into sets α and β and express the linearized mapping as

$$\tilde{f}(\cdot\,;\mathbf{m},\boldsymbol{\Theta}) = \tilde{f}_\alpha(\cdot\,;\mathbf{m},\boldsymbol{\Theta}_\alpha) + \tilde{f}_\beta(\cdot\,;\mathbf{m},\boldsymbol{\Theta}_\beta), \tag{A.27}$$

with

$$\tilde{f}_\alpha(\cdot\,;\mathbf{m},\boldsymbol{\Theta}_\alpha) \doteq \mathcal{J}(\cdot,\mathbf{m})_\alpha(\boldsymbol{\Theta}_\alpha - \mathbf{m}_\alpha), \tag{A.28}$$

and

$$\tilde{f}_\beta(\cdot\,;\mathbf{m},\boldsymbol{\Theta}_\beta) \doteq f(\cdot\,;\mathbf{m}) + \mathcal{J}(\cdot,\mathbf{m})_\beta(\boldsymbol{\Theta}_\beta - \mathbf{m}_\beta), \tag{A.29}$$

where $\mathcal{J}(\cdot,\mathbf{m})_\alpha$ and $\mathcal{J}(\cdot,\mathbf{m})_\beta$ are the columns of the Jacobian matrix corresponding to the sets of parameters α and β, respectively, and $\boldsymbol{\Theta}_\alpha$ and $\boldsymbol{\Theta}_\beta$ are the corresponding random parameter vectors.

Noting that Equation (A.27) expresses $\tilde{f}$ as a sum of (affine transformations of) random variables, we can use the fact that for independent Gaussian random variables $\mathbf{X}$ and $\mathbf{Y}$, the distribution $h_{\mathbf{Z}}$ of $\mathbf{Z} = \mathbf{X} + \mathbf{Y}$ is equal to the convolution of the distributions $h_{\mathbf{X}}$ and $h_{\mathbf{Y}}$ to obtain an approximation to $\tilde{f}$. In particular, for $\mathbf{Z} = \mathbf{X} + \mathbf{Y}$,

$$f_{\mathbf{Z}}(\mathbf{z}) = \int_{-\infty}^{\infty} f_{\mathbf{Y}}(\mathbf{z}-\mathbf{x}) f_{\mathbf{X}}(\mathbf{x})\,\mathrm{d}\mathbf{x}. \tag{A.30}$$

Letting $\mathbf{X} = \tilde{f}_\alpha(\mathbf{X};\mathbf{m},\boldsymbol{\Theta}_\alpha)$, $\mathbf{Y} = \tilde{f}_\beta(\mathbf{X};\mathbf{m},\boldsymbol{\Theta}_\beta)$, and $\mathbf{X} = \tilde{f}(\mathbf{X};\boldsymbol{\Theta})$, we can write

$$\begin{aligned}
&\tilde{g}_{\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\Theta})}(\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\theta})) \\
&= \int_{-\infty}^{\infty} \tilde{g}_{\tilde{f}_\beta(\mathbf{X};\mathbf{m},\boldsymbol{\Theta}_\beta)}(\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\theta}) - \tilde{f}_\alpha(\mathbf{X};\mathbf{m},\boldsymbol{\theta}_\alpha))\tilde{g}_{\tilde{f}_\alpha(\mathbf{X};\mathbf{m},\boldsymbol{\Theta}_\alpha)}(\tilde{f}_\alpha(\mathbf{X};\mathbf{m},\boldsymbol{\theta}_\alpha))\,\mathrm{d}\mathbf{x}, \\
&= \int_{-\infty}^{\infty} \mathcal{N}(\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\theta})\,;\boldsymbol{\mu}(\mathbf{X},\mathbf{m},\boldsymbol{\theta}_\alpha,\tilde{f}_\alpha),\boldsymbol{\Sigma}(\mathbf{X},\mathbf{m},\boldsymbol{\theta}_\beta,\mathbf{S}_\beta))\tilde{g}_{\tilde{f}_\alpha(\mathbf{X};\mathbf{m},\boldsymbol{\Theta}_\alpha)}(\tilde{f}_\alpha(\mathbf{X};\mathbf{m},\boldsymbol{\theta}_\alpha))\,\mathrm{d}\mathbf{x},
\end{aligned} \tag{A.31}$$

with

$$\boldsymbol{\mu}(\mathbf{X},\mathbf{m},\tilde{f}_\alpha(\mathbf{X};\mathbf{m},\boldsymbol{\theta}_\alpha)) = f(\mathbf{X};\mathbf{m}) + \tilde{f}_\alpha(\mathbf{X};\mathbf{m},\boldsymbol{\theta}_\alpha) \tag{A.32}$$

and

$$\boldsymbol{\Sigma}(\mathbf{X},\mathbf{m},\boldsymbol{\theta}_\beta,\mathbf{S}_\beta) = \mathcal{J}(\mathbf{X},\mathbf{m})_\beta \mathbf{S}_\beta \mathcal{J}(\mathbf{X}',\mathbf{m})_\beta{}^\top, \tag{A.33}$$

where we have used the fact that for a Gaussian distribution with mean m and covariance S, $\mathcal{N}(z-y; m, S) = \mathcal{N}(z; m+y, S)$. We can then approximate the probability density function $\tilde{g}_{\tilde{f}(\mathbf{X};\mathbf{m},\mathbf{\Theta})}(\tilde{f}(\mathbf{X};\boldsymbol{\theta}))$ via the Monte Carlo estimator

$$\hat{\tilde{g}}_{\tilde{f}(\mathbf{X};\mathbf{m},\mathbf{\Theta})}(\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\theta})) = \frac{1}{R}\sum_{j=1}^{R}\mathcal{N}(\tilde{f}(\mathbf{X};\mathbf{m},\boldsymbol{\theta})\,;\boldsymbol{\mu}(\mathbf{X},\mathbf{m},\tilde{f}_\alpha(\mathbf{X};\mathbf{m},\boldsymbol{\theta}_\alpha)^{(j)}),\mathbf{\Sigma}(\mathbf{X},\mathbf{m},\boldsymbol{\theta}_\beta,\mathbf{S}_\beta)) \tag{A.34}$$

with $\tilde{f}_\alpha(\mathbf{X};\mathbf{m},\boldsymbol{\theta}_\alpha)^{(j)} \sim \tilde{g}_{\tilde{f}_\alpha(\mathbf{X};\mathbf{m},\mathbf{\Theta}_\alpha)}$. Finally, we can express $\hat{\tilde{g}}_{\tilde{f}(\mathbf{X};\mathbf{m},\mathbf{\Theta})}(\tilde{f}(\mathbf{X};\boldsymbol{\theta}))$ equivalently as

$$\hat{\tilde{g}}_{\tilde{f}(\mathbf{X};\mathbf{m},\mathbf{\Theta})} = \mathcal{N}\Big(f(\mathbf{X};\mathbf{m}) + \frac{1}{R}\sum_{j=1}^{R}\tilde{f}_\alpha(\mathbf{X};\mathbf{m},\mathbf{\Theta}_\alpha)^{(j)}, \mathcal{J}(\mathbf{X},\mathbf{m})_\beta\mathbf{S}_\beta\mathcal{J}(\mathbf{X}',\mathbf{m})_\beta{}^\top\Big), \tag{A.35}$$

where $g_{\mathbf{\Theta}_\beta} = \mathcal{N}(\mathbf{m}_\beta, \mathbf{S}_\beta)$ and samples $\tilde{f}_\alpha(\mathbf{X};\mathbf{m},\mathbf{\Theta}_\alpha)^{(j)}$ are obtained by sampling parameters from the distribution $g_{\mathbf{\Theta}_\alpha} = \mathcal{N}(\mathbf{m}_\alpha, \mathbf{S}_\alpha)$. This concludes the proof. □

Proposition 3 (Covariance of Distribution over Linearized Mapping Induced by Distribution over Parameters with Mean Zero)**.** *Let $\sigma : \mathbb{R}^m \to \mathbb{R}^n$ be an elementwise mapping with $\sigma(0) = 0$, let $f : \mathcal{X} \times \mathbb{R}^P \to \mathbb{R}^Q$ be a mapping that can be decomposed as*

$$f(\mathbf{x};\mathbf{\Theta}_{W,b}) = \sigma(\mathbf{\Phi}(\mathbf{x};\mathbf{\Theta}^{\mathbf{\Phi}}_{W,b}))^\top\mathbf{\Theta}^{\text{linear}}_W + \mathbf{\Theta}^{\text{linear}}_b, \tag{A.36}$$

where $\mathbf{\Phi} : \mathcal{X} \times \mathbb{R}^{P'} \to \mathbb{R}^{P''}$ and

$$\mathbf{\Theta}_{W,b} = \text{vec}(\{\mathbf{\Theta}^{\mathbf{\Phi}}_{W,b}, \mathbf{\Theta}^{\text{linear}}_W, \mathbf{\Theta}^{\text{linear}}_b\}) \in \mathbb{R}^P, \quad \mathbf{\Theta}^{\mathbf{\Phi}}_{W,b} \in \mathbb{R}^{P'}, \quad \mathbf{\Theta}^{\text{linear}}_W \in \mathbb{R}^{P''\times Q}, \quad \mathbf{\Theta}^{\text{linear}}_b \in \mathbb{R}^Q,$$

and let the Jacobian of f evaluated at $\mathbf{m} \in \mathbf{R}^P$ be denoted by $\mathcal{J}(\cdot,\mathbf{m}) \doteq \frac{\partial f(\cdot\,;\mathbf{\Theta})}{\partial\mathbf{\Theta}}|_{\mathbf{\Theta}=\mathbf{m}}$. Then the covariance matrix of f evaluated $x \in \mathbb{R}^D$ and $\mathbf{m} = \mathbf{0}$ induced by a distribution over the network parameters with variance $\mathbf{\Sigma}$ is equal to the identity operator, that is, for output dimension j

$$\mathcal{J}(\mathbf{x},\mathbf{0})\Sigma_j\mathcal{J}(\mathbf{x}',\mathbf{0})^\top = \mathbf{\Sigma}^{\text{linear}}_{b,j}, \tag{A.37}$$

where $\mathbf{\Sigma}^{\text{linear}}_{b,j}$ is a variance parameter associated with the final-layer bias for output dimension j.

Proof. The mapping $\mathcal{J}(\mathbf{x},\mathbf{m})$ gives the change in f at evaluation points x and $\mathbf{m}$ for infinitesimally small changes in each $\mathbf{m}_i$. If $\mathbf{m} = \mathbf{0}$, an increase of $\boldsymbol{\epsilon}$ in any $\mathbf{m}_i$ will not change the image of f at x, since for any $\mathbf{m}_i = \boldsymbol{\epsilon}$ there exists some $\mathbf{m}_j = \mathbf{0}$ with $i \neq j$ that is multiplied with $h(\mathbf{m}_i)$ (for any mapping h); the only exception being the bias term $\mathbf{\Theta}^{\text{linear}}_b$. Since $\frac{\partial\mathbf{\Theta}^{\text{linear}}_b}{\partial\mathbf{\Theta}^{\text{linear}}_b} = \mathbb{I}_Q$,

$$[\mathcal{J}(\mathbf{x},\mathbf{0})]_k = \left[\mathbf{0}^\top_{P'+P''}, \left[\frac{\partial\mathbf{\Theta}^{\text{linear}}_b}{\partial\mathbf{\Theta}^{\text{linear}}_b}\right]_k\right]^\top, \tag{A.38}$$

where $1 \leq k \leq Q$, $[\mathcal{J}(\mathbf{x},\mathbf{0})]_k$ is the kth row of $\mathcal{J}(\mathbf{x},\mathbf{0})$, and $\left[\frac{\partial\mathbf{\Theta}^{\text{linear}}_b}{\partial\mathbf{\Theta}^{\text{linear}}_b}\right]_k$ is the kth row of $\frac{\partial\mathbf{\Theta}^{\text{linear}}_b}{\partial\mathbf{\Theta}^{\text{linear}}_b} = \mathbb{I}_Q$, we get

$$[\mathcal{J}(\mathbf{x},\mathbf{0})]_k = \left[\mathbf{0}^\top_{P'+P''}, \left[\frac{\partial\mathbf{\Theta}^{\text{linear}}_b}{\partial\mathbf{\Theta}^{\text{linear}}_b}\right]_k\right] = \left[\mathbf{0}^\top_{P'+P''}, \mathbf{0}^\top_{k-1}, 1, \mathbf{0}^\top_{Q-k-1}\right], \tag{A.39}$$

and hence, for output dimension j,

$$\mathcal{J}(\mathbf{x},\mathbf{0})\mathbf{\Sigma}_j\mathcal{J}(\mathbf{x}',\mathbf{0})^\top = \Sigma^{\text{linear}}_{b,j}, \tag{A.40}$$

where $\mathbf{\Sigma}^{\text{linear}}_{b,j}$ is a variance parameter associated with the final-layer bias for output dimension j. For $\mathbf{X}$ containing n evaluation points, we get

$$\mathcal{J}(\mathbf{X},\mathbf{0})\mathbf{\Sigma}_j\mathcal{J}(\mathbf{X}',\mathbf{0})^\top = \Sigma^{\text{linear}}_{b,j}\mathbb{I}_n. \tag{A.41}$$

□

Appendix B Further Empirical Results

B.1 Tabular Results for Diabetic Retinopathy Diagnosis Tasks

The results below were reproduced from Band et al. [2021] using the RETINA benchmark.

Table 3: Country Shift. Prediction and uncertainty quality of baseline methods in terms of the area under the receiver operating characteristic curve (AUC) and classification accuracy, as a function of the proportion of data referred to a medical expert. All methods are tuned on in-domain validation AUC, and ensembles have $K = 3$ constituent models (true for all subsequent tables unless specified otherwise). On in-domain data, MC DROPOUT performs best across all thresholds. On distributionally shifted data, no method consistently performs best.

	No Referral		50% Data Referred		70% Data Referred	
Method	**AUC (%) ↑**	**Accuracy (%) ↑**	**AUC (%) ↑**	**Accuracy (%) ↑**	**AUC (%) ↑**	**Accuracy ↑**
EyePACS Dataset (In-Domain)						
MAP (Deterministic)	87.4±1.3	88.6±0.7	91.1±1.8	95.9±0.4	94.9±1.1	96.5±0.3
MFVI	83.3±0.2	85.7±0.1	85.5±0.7	94.5±0.1	88.2±0.7	95.9±0.1
RADIAL-MFVI	83.2±0.5	74.2±5.0	88.9±0.9	81.8±6.0	91.2±1.3	83.8±5.5
FSVI	88.5±0.1	89.8±0.0	91.0±0.4	96.4±0.0	94.3±0.3	97.2±0.1
MC DROPOUT	91.4±0.2	90.9±0.1	95.3±0.2	97.4±0.1	97.4±0.1	98.1±0.0
RANK-1	85.6±1.4	87.7±0.8	87.1±2.3	95.3±0.5	90.9±2.0	96.4±0.4
DEEP ENSEMBLE	90.3±0.2	90.3±0.3	91.7±0.6	97.2±0.0	95.0±0.5	97.9±0.0
MFVI ENSEMBLE	85.4±0.0	87.8±0.0	86.3±0.4	95.4±0.0	89.2±0.4	96.7±0.1
RADIAL-MFVI ENSEMBLE	84.9±0.1	74.2±1.5	91.4±0.2	83.4±1.7	93.3±0.3	85.9±1.6
FSVI ENSEMBLE	90.3±0.1	90.6±0.0	92.1±0.2	97.1±0.0	95.2±0.2	97.8±0.1
MC DROPOUT ENSEMBLE	**92.5±0.0**	**91.6±0.0**	**95.8±0.1**	**97.8±0.0**	**97.7±0.1**	**98.4±0.0**
RANK-1 ENSEMBLE	89.5±0.8	89.3±0.4	88.5±1.3	96.9±0.3	91.6±1.2	97.6±0.3
APTOS 2019 Dataset (Population Shift)						
MAP (Deterministic)	92.2±0.2	86.2±0.6	80.1±3.6	87.6±1.5	55.4±4.3	85.4±1.2
MFVI	91.4±0.2	84.1±0.3	93.8±0.4	92.1±0.5	93.0±0.6	92.7±0.5
RADIAL-MFVI	90.7±0.7	71.8±4.6	82.0±2.5	81.5±2.7	66.4±2.1	85.9±1.0
FSVI	94.1±0.1	87.6±0.5	90.6±0.9	90.7±0.7	77.2±4.6	89.8±0.3
MC DROPOUT	94.0±0.2	86.8±0.2	87.4±0.3	88.1±0.2	65.3±1.7	88.2±0.4
RANK-1	92.5±0.3	86.2±0.5	90.1±2.5	91.4±1.1	75.1±7.8	89.5±1.5
DEEP ENSEMBLE	94.2±0.2	87.5±0.1	91.2±1.9	92.4±0.9	67.4±7.3	90.1±1.2
MFVI ENSEMBLE	93.2±0.1	87.0±0.2	**94.9±0.3**	**93.7±0.3**	**94.2±0.3**	**94.0±0.3**
RADIAL-MFVI ENSEMBLE	91.8±0.2	69.0±1.9	78.6±0.6	79.8±0.9	60.9±0.3	86.7±0.2
FSVI ENSEMBLE	**94.6±0.1**	**88.9±0.2**	90.7±0.5	91.1±0.6	74.1±3.4	89.8±0.2
MC DROPOUT ENSEMBLE	94.1±0.1	87.6±0.1	86.8±0.2	88.0±0.2	62.3±0.4	87.7±0.2
RANK-1 ENSEMBLE	94.1±0.2	88.3±0.2	**94.9±0.4**	93.5±0.3	92.4±1.5	93.8±0.3

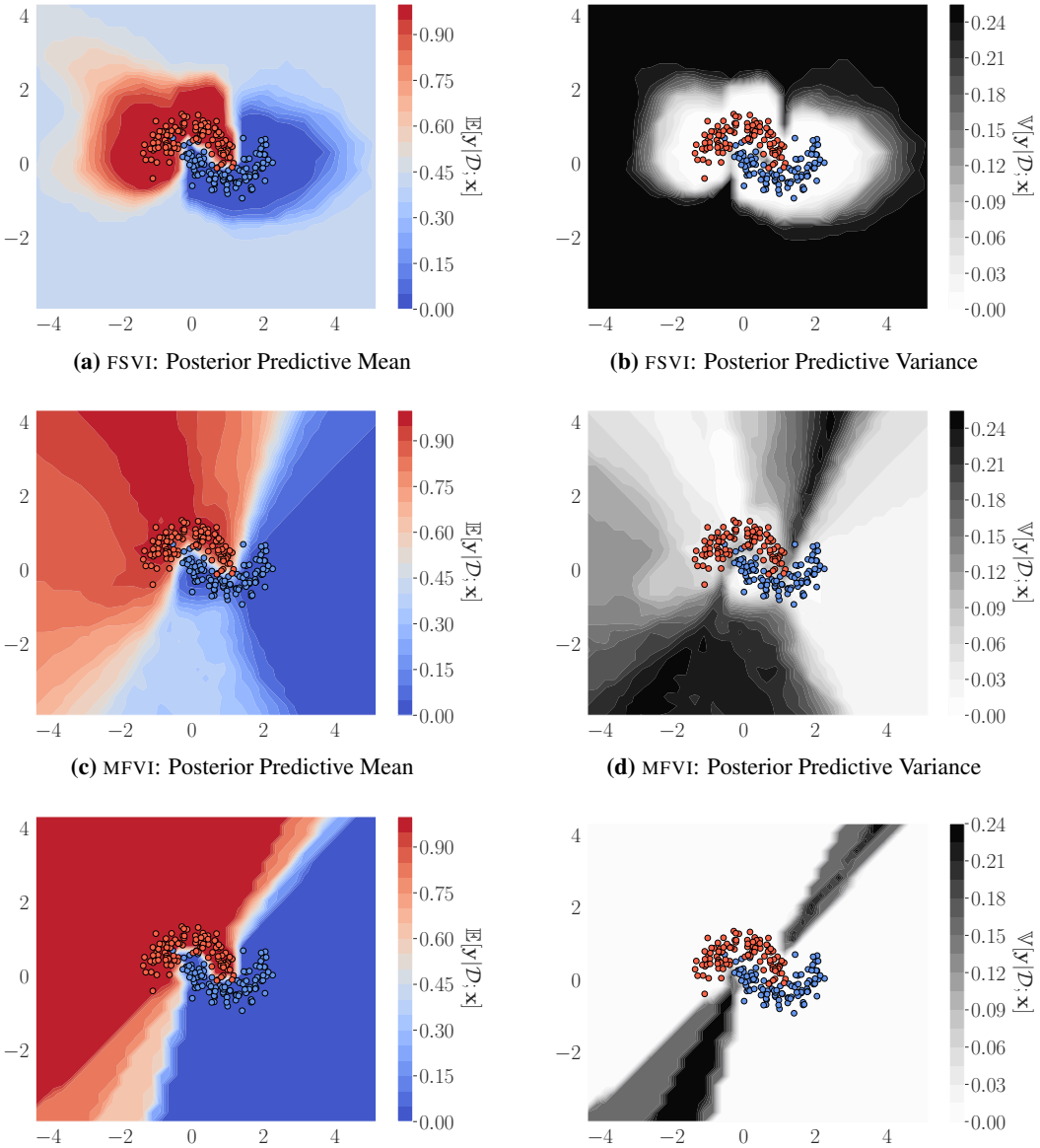
B.2 UCI Regression

Table 4: This table compares the predictive performance between the method proposed in this paper and the method proposed by Sun et al. [2019] on six datasets from the UCI database. We followed the same training protocol as Sun et al. [2019] and used the code provided by the authors to load and process the data. The same network architecture was used (one hidden layer with 50 hidden units). We report the results for the best set of hyperparameters, computed over ten random seeds. Lower RMSE and higher log-likelihood are better. Best results are shaded in gray. The first five rows are small-scale UCI experiments, and the sixth row ("Protein") is a larger-scale experiment (45,740 data points).

	RMSE		Log-Likelihood	
	Sun et al. [2019]	**Ours**	Sun et al. [2019]	**Ours**
Boston	$\mathbf{2.378 \pm 0.104}$	3.632 ± 0.515	$\mathbf{-2.301 \pm 0.038}$	-3.150 ± 0.495
Concrete	4.935 ± 0.180	$\mathbf{4.177 \pm 0.443}$	-3.096 ± 0.016	$\mathbf{-2.855 \pm 0.116}$
Energy	0.412 ± 0.017	$\mathbf{0.409 \pm 0.060}$	-0.684 ± 0.020	$\mathbf{-0.539 \pm 0.138}$
Wine	0.673 ± 0.014	$\mathbf{0.615 \pm 0.033}$	-1.040 ± 0.013	$\mathbf{-0.959 \pm 0.034}$
Yacht	0.607 ± 0.068	$\mathbf{0.514 \pm 0.242}$	-1.033 ± 0.033	$\mathbf{-0.888 \pm 0.334}$
Protein	4.326 ± 0.019	$\mathbf{4.248 \pm 0.043}$	-2.892 ± 0.004	$\mathbf{-2.866 \pm 0.009}$

Appendix C Illustrative Examples

C.1 Two Moons Classification Problem



(a) FSVI: Posterior Predictive Mean

(b) FSVI: Posterior Predictive Variance

(c) MFVI: Posterior Predictive Mean

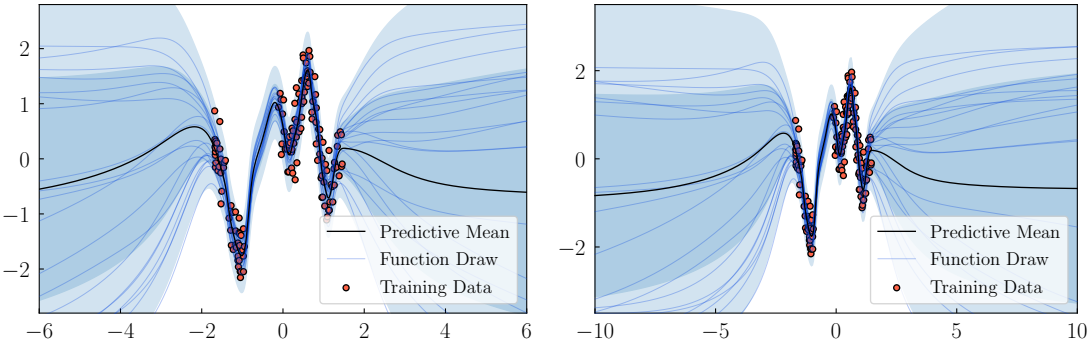
(d) MFVI: Posterior Predictive Variance

(e) MAP Ensemble: Predictive Mean

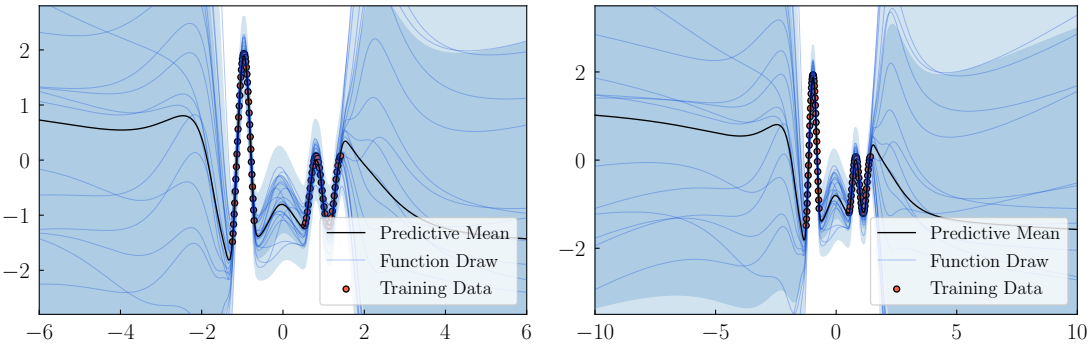
(f) MAP Ensemble: Predictive Variance

Figure 5: Binary classification on the *Two Moons* dataset. The plots show the posterior predictive mean and variance of a BNN trained via FSVI (Figure 5a and Figure 5b), of a BNN trained via MFVI (Figure 5c and Figure 5d), and an ensemble of MAP models (Figure 5e and Figure 5f). The predictive means represent the expected class probabilities and the predictive variance the model's epistemic uncertainty over the class probabilities. With FSVI, the predictive distribution is able to faithfully capture the geometry of the data manifold and exhibits high uncertainty over the class probabilities in areas of the data space of which the data is not informative. In contrast, neither MFVI, nor MAP ensembles are unable to accurately capture the geometry of the data manifold only exhibit high uncertainty around the decision boundary.

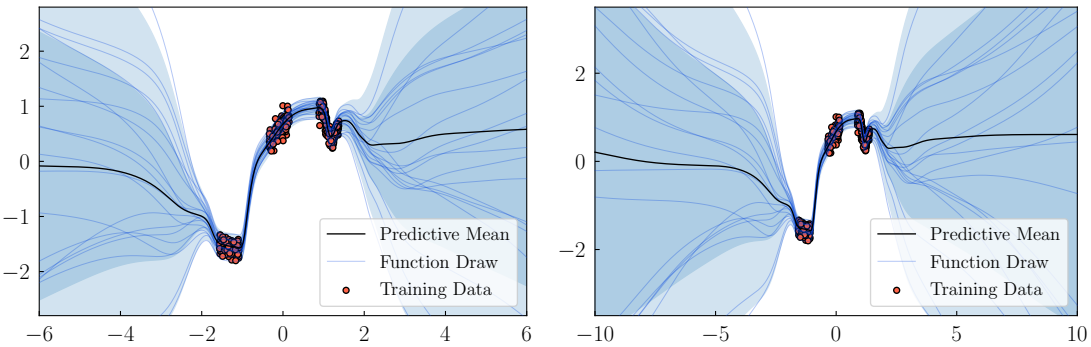
C.2 Synthetic 1D Regression Problems



(a) "Snelson" Dataset (Snelson and Ghahramani [2006])



(b) "OAT-1D" Dataset (van Amersfoort et al. [2021])



(c) "Subspace Inference" Dataset (Izmailov et al. [2020])

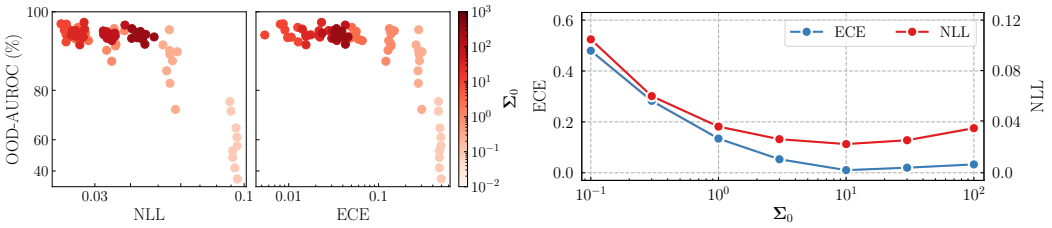
Figure 6: 1D Regression with FSVI on a selection of datasets used to demonstrate desirable predictive uncertainty estimates in prior works. The left column is zoomed in.

Appendix D Sensitivity Studies

We performed a comprehensive set of sensitivity studies to elucidate how different model and variational parameters affect the resulting predictive distributions. Specifically, we used a validation holdout set (10% of the training data) to assess the effect of the prior covariance, the number of context points used per gradient steps, the context distribution, and the number of Monte Carlo samples for evaluating the expected log-likelihood on in- and out-of-distribution performance metrics. All results below are obtained from ten random seeds.

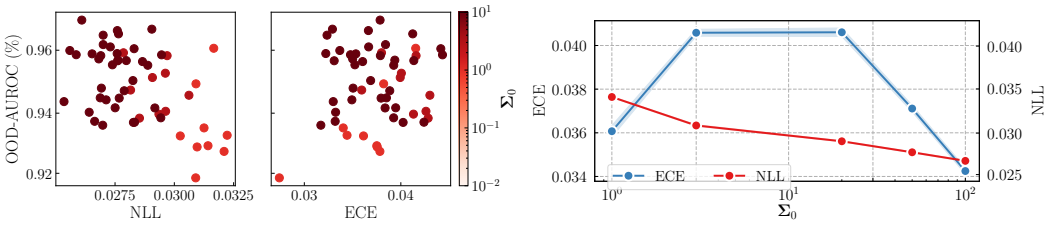
D.1 Ablation Study on the Effect of Different Function-Space Priors

To better understand FSVI, we investigate what hyperparameter choice leads to good predictive performance and reliable out-of-distribution predictive uncertainty. Figure 7 and Figure 8 show plots that demonstrate the link between different choices of prior variance and the resulting OOD-AUROC, test ECE, and test log-likelihood for FashionMNIST. As can be seen in Figure 7b, test ECE is lowest for a prior variance of $\Sigma_0 = 10$, while Figure 7a show that OOD-AUROC, test ECE, and test log-likelihood are highly negatively correlated. This insight is useful, since it means that in real-world settings, where there are no real out-of-distribution validation sets, one can choose the prior variance that minimize test ECE and test negative log-likelihood. We follow this approach to select the hyperparameters for Table 1 and Table 2.



(a) Effect of Prior Variance Σ_0 **(b)** ECE and NLL vs. Σ_0

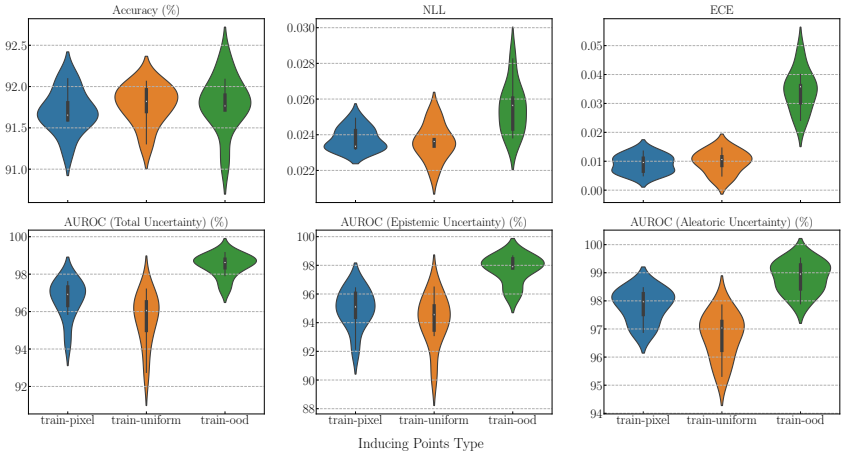
Figure 7: For BNNs trained via FSVI on FashionMNIST, the figures show the relationship between OOD-AUROC (on MNIST), the negative log-likelihood (NLL), expected calibration error (ECE), and the prior variance. OOD-AUROC is very negatively correlated with both NLL and ECE. For further results, see Appendix B.



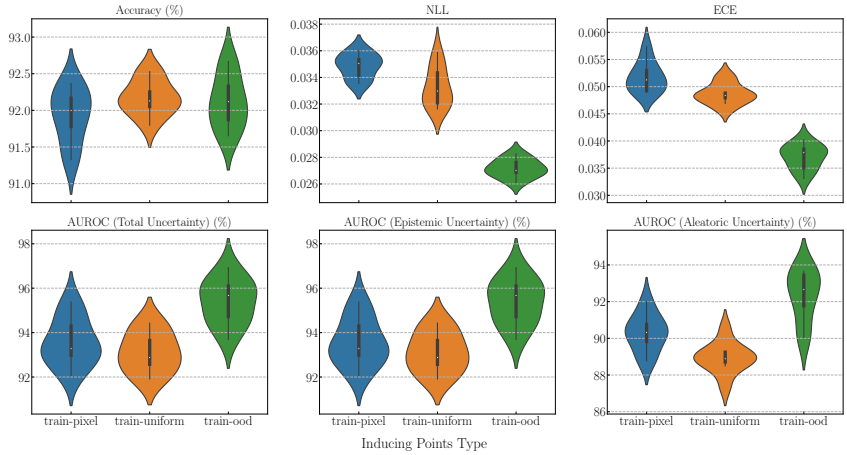
(a) Effect of Prior Variance Σ_0 **(b)** ECE and NLL vs. Σ_0

Figure 8: For BNNs trained via FSVI on CIFAR-10, the figures show the relationship between OOD-AUROC (on SVHN), the negative log-likelihood (NLL), expected calibration error (ECE), and the prior variance. OOD-AUROC is negatively correlated with NLL as the prior variance changes (i.e., the larger the prior variance, the lower the NLL and the higher the OOD-AUROC), and there appears to be a weak positive correlation between OOD-AUROC and ECE as the prior variance changes (i.e., after some variance threshold, the larger the prior variance, the lower are both NLL and ECE). For further results, see Appendix B.

D.2 Context Distribution



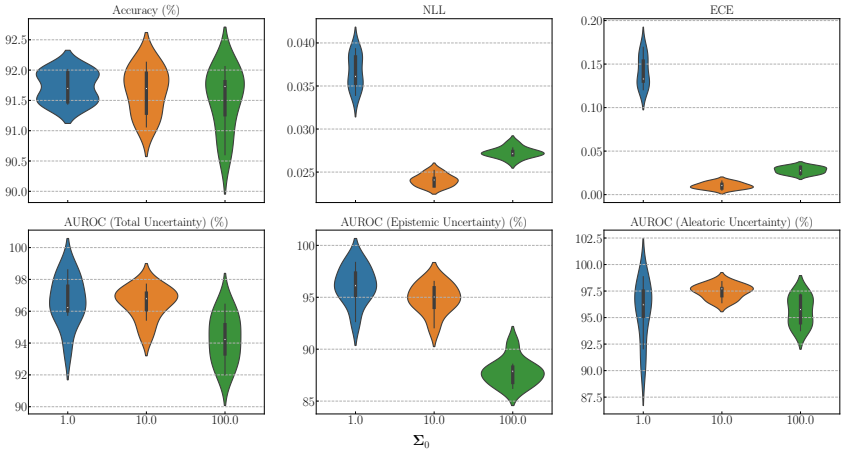
(a) FashionMNIST



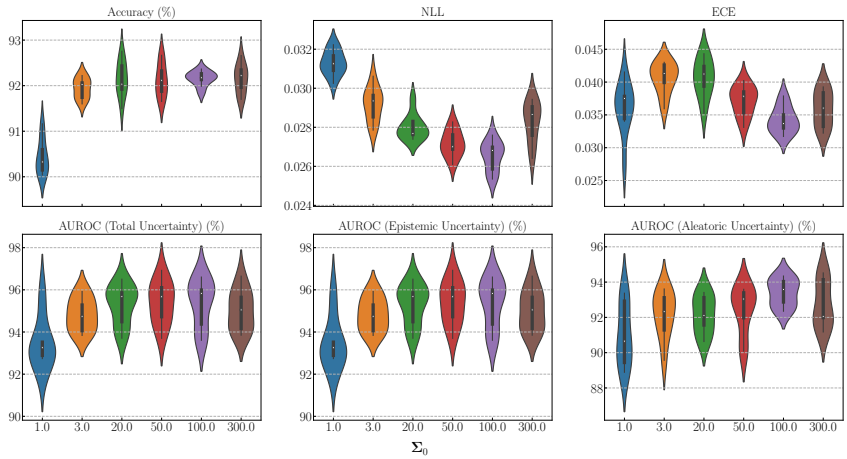
(b) CIFAR-10

Figure 9: Effect of the context distribution on in- and out-of-distribution evaluation metrics. A prior variance of 10 was used for FashionMNIST and a prior variance of 50 was used for CIFAR-10. For all experiments, three sets of 10 context points each were sampled to estimate the supremum at each gradient step. 5 Monte Carlo samples were used to evaluate the expected log-likelihood. Out-of-distribution detection is performed on MNIST for models trained on FashionMNIST and on SVHN for models trained on CIFAR-10.

D.3 Prior Variance



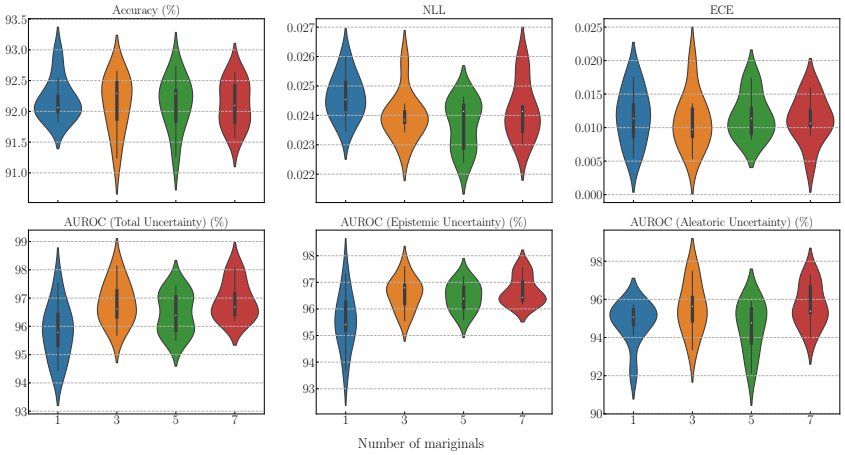
(a) FashionMNIST



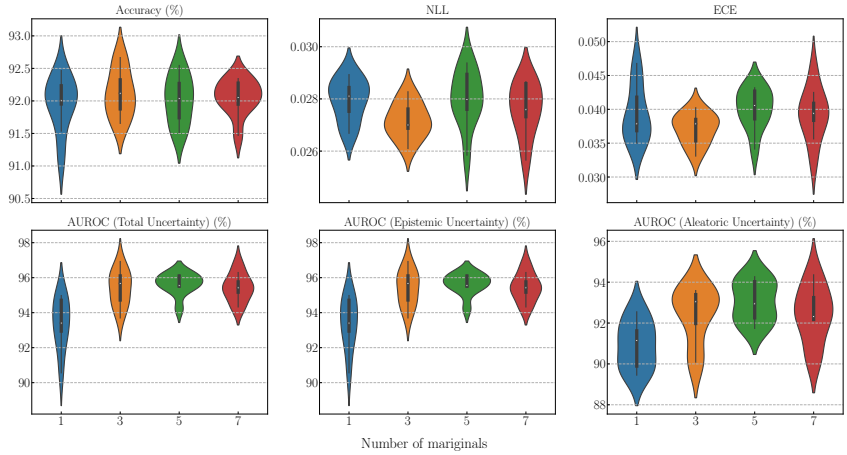
(b) CIFAR-10

Figure 10: Effect of prior variance over parameters $\mathbf{\Sigma}_0$ on in- and out-of-distribution evaluation metrics. For all experiments, three sets of 10 context points each were sampled to estimate the supremum at each gradient step. 5 Monte Carlo samples were used to evaluate the expected log-likelihood. Out-of-distribution detection is performed on MNIST for models trained on FashionMNIST and on SVHN for models trained on CIFAR-10.

D.4 Number of Sets of Context Points



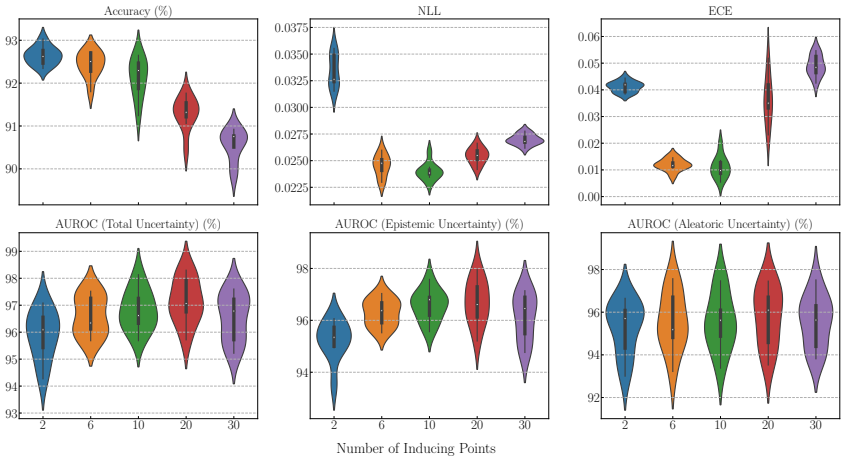
(a) FashionMNIST



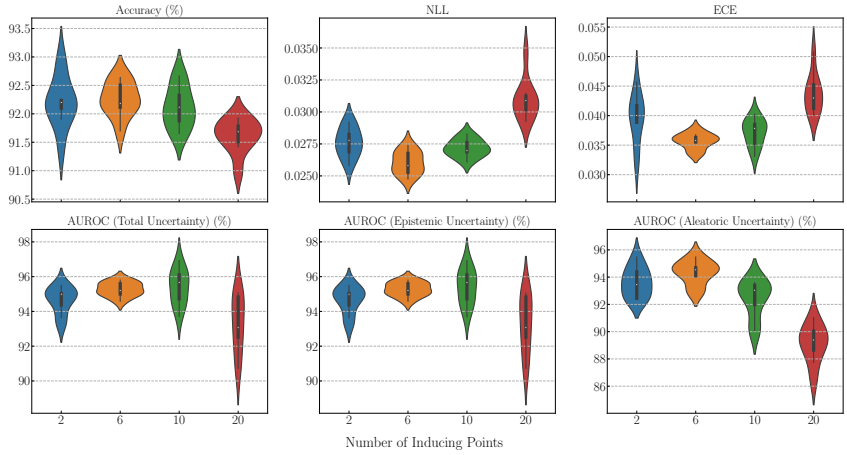
(b) CIFAR-10

Figure 11: Effect of the number of context points sets from which the supremum is estimated on in- and out-of-distribution evaluation metrics. A prior variance of 10 was used for FashionMNIST and a prior variance of 50 was used for CIFAR-10. For all experiments, each set $\mathbf{X}_C^{(i)}$ from which the supremum is estimated contains 10 context points at each gradient step. 5 Monte Carlo samples were used to evaluate the expected log-likelihood. Out-of-distribution detection is performed on MNIST for models trained on FashionMNIST and on SVHN for models trained on CIFAR-10.

D.5 Number of Context Points



(a) FashionMNIST



(b) CIFAR-10

Figure 12: Effect of the number of context points per set $\mathbf{X}_{\mathcal{C}}^{(i)}$ on in- and out-of-distribution evaluation metrics. A prior variance of 10 was used for FashionMNIST and a prior variance of 50 was used for CIFAR-10. For all experiments, three sets of varying numbers of context points each were sampled to estimate the supremum at each gradient step. 5 Monte Carlo samples were used to evaluate the expected log-likelihood. Out-of-distribution detection is performed on MNIST for models trained on FashionMNIST and on SVHN for models trained on CIFAR-10.

Appendix E Implementation, Training, and Evaluation Details

E.1 Hyperparameter Selection Protocol

For FSVI, we used a holdout validation set (10% of the training set) to conduct a hyperparameter search over the prior variance, the number of context points used to evaluate the KL divergence, the context distribution, and the number of Monte Carlo samples used to evaluate the expected log-likelihood. The full results can be found in Appendix D. We selected the set of hyperparameters that yielded the lowest validation log-likelihood for all experiments. We state the hyperparameters selected for the different datasets below. We used 5 Monte Carlo samples and 10 context points per gradient step for all of our experiments.

For other methods, we used a holdout validation set of the same size and selected the best-performing hyperparameters. We used implementations provided by the authors of MFVI (radial) and SWAG. All other methods were implemented from scratch unless stated otherwise.

E.2 FashionMNIST vs. MNIST/NotMNIST

We train all model on the FashionMNIST dataset and evaluate the models' predictive uncertainty performance on out-of-distribution data on the MNIST dataset. Both datasets consist of images of size 28×28 pixels. The FashionMNIST dataset is normalized to have zero mean and a standard deviation of one. The MNIST dataset is normalized with the same transformation, that is, using the same mean and standard deviation used for the in-distribution data. We chose FashionMNIST/MNIST instead of MNIST/notMNIST because the latter is notably easier than the former.

In this experiment, a network architecture with two convolutional layers of 32 and 64 3×3 filters and a fully-connected final layer of 128 hidden units is used. A max pooling operation is placed after each convolutional layer and ReLU activations are used. We do not use batch normalization. All models are trained for 30 epochs with a mini-batch size of 200 and using the Adam optimizer with a learning rate 5×10^{-4}.

For FSVI with $p_{\mathbf{X}_c}$ =random monochrome, we used a prior variance of $\mathbf{\Sigma}_0 = 10$ and sampled 50% of the context points for each gradient step from the mini-batch and the other 50% according to the method described in Appendix E.8. For FSVI with $p_{\mathbf{X}_c}$= KNIST, we used the KMNIST dataset.

E.3 CIFAR-10 vs. SVHN

We train all model on the CIFAR-10 dataset and evaluate the models' predictive uncertainty performance on out-of-distribution data on the SVHN dataset. Both datasets consist of images of size $32 \times 32 \times 3$, with RBG channels. The CIFAR-10 dataset is normalized to have zero mean and a standard deviation of one. The SVHN dataset is normalized with the same transformation, that is, using the same mean and standard deviation used for the in-distribution data. The training data is augmented with random horizontal flips (with a probability of 0.5) and random crops (4 zero pixels on all sides).

In this experiment, a standard ResNet-18 network architecture is used. We do not use batch normalization. All models are trained for 50 epochs and using the Adam optimizer with a mini-batch size of 128 and a learning rate 5×10^{-4}.

For FSVI with $p_{\mathbf{X}_c}$ =random monochrome, we used a prior variance of $\mathbf{\Sigma}_0 = 0.1$ and sampled 100% of the context points for each gradient step from the mini-batch and the other 50% according to the method described in Appendix E.8. For FSVI with $p_{\mathbf{X}_c}$= CIFAR-100, we used the CIFAR-100 dataset.

E.4 Diabetic Retinopathy Diagnosis

Prediction and Expert Referral. In real-world settings where the evaluation data may be sampled from a shifted distribution, incorrect predictions may become increasingly likely. To account for that possibility, predictive uncertainty estimates can be used to identify datapoints where the likelihood of an incorrect prediction is particularly high and refer them for further review. We consider a corresponding selective prediction task, where the predictive performance of a given model is evaluated for varying expert referral rates. That is, for a given referral rate of $\gamma \in [0, 1]$, a model's

predictive uncertainty is used to identify the γ proportion of images in the evaluation set for which the model's predictions are most uncertain. Those images are referred to a medical professional for further review, and the model is assessed on its predictions on the remaining $(1-\gamma)$ proportion of images. By repeating this process for all possible referral rates and assessing the model's predictive performance on the retained images, we estimate how reliable it would be in a safety-critical downstream task, where predictive uncertainty estimates are used in conjunction with human expertise to avoid harmful predictions. Importantly, selective prediction tolerates out-of-distribution examples. For example, even if unfamiliar features appear in certain images, a model with reliable uncertainty estimates will perform better in selective prediction by assigning these images high epistemic (and predictive) uncertainty, therefore referring them to an expert at a lower γ.

For all methods, experiments are performed using a ResNet-50 network architecture. Training and evaluation scripts as well as model checkpoints can be found at

`github.com/google/uncertainty-baselines/.../diabetic_retinopathy_detection`.

E.5 Two Moons

In this experiment, a network architecture with with two fully-connected layer with 30 hidden units each is used. We train all models with a learning rate of 10^{-3}.

For FSVI, we used a prior variance of $\mathbf{\Sigma}_0 = 10$ and sampled context points randomly from $[-10, 10]^2$.

E.6 1D Regression

In this experiment, we use a model consisting of two fully-connected layers with 100 hidden units each and Tanh activations. We train all models with a learning rate of 10^{-3}.

For FSVI, we used a prior variance of $\mathbf{\Sigma}_0 = 10$ and sampled context points randomly from $[-10, 10]$.

E.7 Further Implementation Details

We use the Adam optimizer with default settings of $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\epsilon = 10^{-8}$ for all experiments. The deterministic neural networks that were used for the ensemble were trained with a weight decay of λ = 1e-1. MFVI (tempered) was trained with a KL scaling factor of 0.1 to obtain a cold posterior.

E.8 Selection of Context Distribution

We estimate the supremum at every gradient step by sampling a set of context points $\mathbf{X}_\mathcal{C}$ from a distribution $p_{\mathbf{X}_\mathcal{C}}$ at every gradient step. For tasks with image inputs, we construct a distribution $p_{\mathbf{X}_\mathcal{C}}$, defined as a uniform distribution over images with monochromatic channels. To generate a sample from this "monochrome images" distribution, we first take all images in the training data, flatten each channel, and stack the flattened image channels into a single vector each. We then draw a random element (i.e., a pixel) from each channel vector and then use these pixels to generate a monochrome image of a given resolution by setting every channel equal to the value of the pixel that was drawn. For regression tasks with a D-dimensional input space, $p_{\mathbf{X}_\mathcal{C}}$ is defined as a uniform distribution with lower and upper bounds set to the empirical lower and upper bounds of the training data. For further details on the effect of different sampling schemes on the posterior predictive distribution's performance, see Appendix B.

E.9 Compute Resources

All experiments were carried out on an Nvidia V-100 GPU with 32GB of memory.