Tractable Function-Space Variational Inference in Bayesian Neural Networks

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

Abstract

The most common approach to inference in Bayesian neural networks is to approximate the posterior distribution over the network parameters. However, explicit inference over the network parameters can make it difficult to incorporate meaningful prior information about the prediction task into the inference process. In this paper, we consider an alternative approach. Taking advantage of the fact that Bayesian neural networks define distributions over functions induced by distributions over parameters, we propose a scalable and tractable method for function-space variational inference. We evaluate the proposed method empirically and show that it leads to competitive predictive accuracy and reliable predictive uncertainty estimation on a range of prediction problems and performs well on safety-critical downstream tasks where reliable uncertainty estimation is essential.

1 Introduction

Machine learning models succeed at an increasingly wide range of narrowly defined tasks, but fail without warning when used on inputs that are meaningfully different from the data they were trained on. 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.

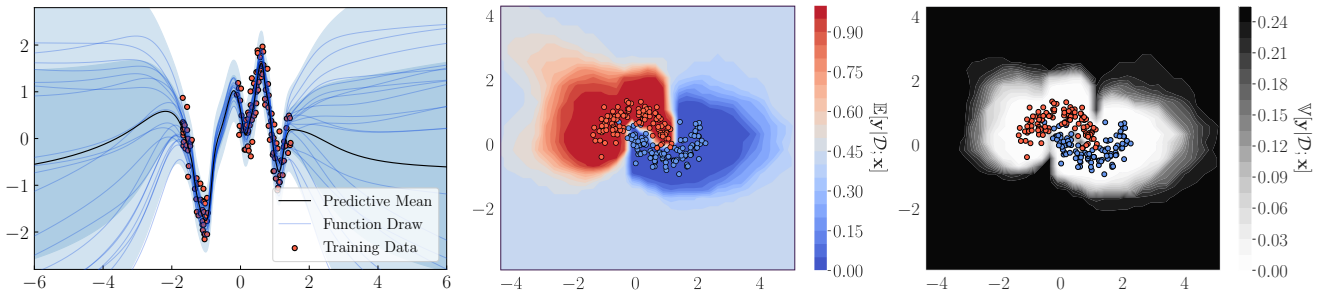
In this paper, we develop a method for obtaining reliable uncertainty estimates in Bayesian neural networks (BNNs). While BNNs have promised to combine the advantages of deep learning and Bayesian inference, existing approaches for approximate inference in large BNNs fall short of this promise and result 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. A reason for this shortcoming is that commonly used parameter-space inference methods make it difficult to define meaningful priors that effectively incorporate prior information about the data into inference.

To avoid this limitation, we derive a variational objective defined explicitly in terms of the distributions over *functions* induced by a variational distribution over parameters. The proposed approach diverges from prior work on function-space variational inference in that it results in a tractable variational objective amenable to stochastic variational inference instead of relying on approximate stochastic gradients of an otherwise intractable objective. The proposed variational objective allows defining priors that explicitly encourage high uncertainty away from the training data as well as priors that contain prior information about the task at hand.

We demonstrate that our function-space variational objective leads to Bayesian neural networks with significantly improved predictive uncertainty estimates compared to a wide array of state-of-the-art Bayesian and non-Bayesian methods. Figure 1 shows examples of predictive distributions obtained with our method on easy-to-visualize synthetic datasets. As can be seen from the figures, the function-space variational objective leads to BNNs with predictive distributions that rival exact posterior distributions of, for example, Gaussian process models.

Contributions. We propose a tractable and scalable approach for performing function-space variational inference in BNNs. The variational method scales to high-dimensional data, allows for the incorporation

[*] Equal contribution.
[†] Corresponding author: tim.rudner@cs.ox.ac.uk.
Preprint (October 31, 2021).



(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 D.

of meaningful prior information about the data into the inference process, and produces reliable predictive uncertainty estimates. We perform an extensive empirical evaluation in which we compare the proposed approach to a wide array of competing methods and show that it consistently results in a high predictive performance and reliable predictive uncertainty estimates, outperforming other methods in terms of test accuracy, uncertainty-based out-of-distribution detection, out-of-distribution confidence, uncertainty-based selective prediction. We also evaluate the proposed method on a continual learning downstream task that requires effective use of prior information and show that it outperforms related methods.

2 Background

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 $\boldsymbol{\Theta} \in \mathbb{R}^P$ and letting the function mapping defined by a neural network architecture be given by $f : \mathcal{X} \times \mathbb{R}^P \to \mathcal{Y}$, we obtain a random function $f(\cdot\,; \boldsymbol{\Theta})$. For a parameter realization $\boldsymbol{\theta}$, we obtain a corresponding function realization, $f(\cdot\,; \boldsymbol{\theta})$. When evaluated at a finite collection of points $\mathbf{X} \in \mathcal{X}$, the stochastic function $f(\cdot\,; \boldsymbol{\Theta})$ turns into a multivariate random variable, $f(\mathbf{X}; \boldsymbol{\Theta})$, and the function $f(\cdot\,; \boldsymbol{\theta})$ turns into a vector $f(\mathbf{X}; \boldsymbol{\Theta})$.

Letting $p(\mathbf{y}_\mathcal{D} \,|\, f(\mathbf{X}_\mathcal{D}; \boldsymbol{\theta}))$ be the likelihood of observing the targets $\mathbf{y}$ under the stochastic function $f(\cdot\,; \boldsymbol{\Theta})$ evaluated at inputs $\mathbf{X}_\mathcal{D}$ and letting $p_{\boldsymbol{\Theta}}$ be a prior distribution over the stochastic network parameters $\boldsymbol{\Theta}$, we can use Bayesian inference for finding the posterior distribution over parameters given the observed data, $p_{\boldsymbol{\Theta}|\mathcal{D}}$ (MacKay, 1992; Neal, 1996). However, since the mapping f is a nonlinear function of the stochastic parameters $\boldsymbol{\Theta}$, exact inference is analytically intractable. Variational inference is an approximate method that seeks to sidestep this intractability by framing posterior inference as a variational optimization problem. Following this approach, we can obtain a Bayesian neural network defined in terms of a variational distribution over parameters $q_{\boldsymbol{\Theta}}$ by solving $\min_{q_{\boldsymbol{\Theta}} \in \mathcal{Q}_{q_\Theta}} \mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\Theta}} \parallel p_{\boldsymbol{\Theta}|\mathcal{D}})$, where $\mathcal{Q}_{q_{\boldsymbol{\Theta}}}$ is a variational family of distributions. 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. (Blundell et al., 2015; Graves, 2011; Hinton and van Camp, 1993; Hoffman et al., 2013; Wainwright and Jordan, 2008). Henceforth, we will refer to BNN inference methods that make these variational assumptions as parameter-space mean-field variational inference (MFVI).

While MFVI is compatible with techniques used in modern deep learning and can be scaled to large networks, prior work has identified several empirical issues with parameter-space MFVI in BNNs. For example, parameter-space MFVI has been shown to suffer from a decreasing signal-to-noise ratio in the expected log-likelihood gradients and requires ad-hoc fixes to stabilize training, such as modifying the variational objective via temperature scaling (Wenzel et al., 2020) and changes to the gradient estimator (Farquhar et al., 2020a).

3 A Function-Space Perspective on Variational Inference in BNNs

In this section, we present a function-space view of variational inference in BNNs and discuss shortcomings of prior approaches to function-space variational inference (FSVI).

By considering a prior distribution over functions, $p_{f(\cdot;\boldsymbol{\Theta})}$, we can explicitly frame Bayesian inference in BNNs as inferring a distribution over *functions*. Specifically, we can express 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 the variational distribution over functions induced by a variational distribution $q_{\boldsymbol{\Theta}}$. As discussed by Burt et al. (2021), this variational objective function is 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 (that is, 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 $\boldsymbol{\Theta}$ if and only if $q_{\boldsymbol{\Theta}}$ is 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 $\mathbf{y}$ and a suitably defined prior distribution over functions, we can express the variational problem above equivalently as the well-defined maximization problem $\max_{q_{\boldsymbol{\Theta}} \in \mathcal{Q}_{\boldsymbol{\theta}}} \mathcal{F}(q_{\boldsymbol{\Theta}}, \mathbf{X}_{\mathcal{D}}, \mathbf{y})$ with

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

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, if $q_{f(\cdot;\boldsymbol{\Theta})}$ and $p_{f(\cdot;\boldsymbol{\Theta})}$ are variational and prior distributions over functions, respectively, evaluating the KL divergence in Equation (2) is intractable. 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, $\mathbf{X}_{\mathcal{I}}$, resulting in the objective function

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

However, this objective function is still challenging to optimize in practice: The supremum cannot be obtained analytically, Sun et al. (2019) reported that treating the supremum as an adversarial optimization objective was challenging in practice, and the KL divergence term itself is intractable analytically and difficult to estimate—even for finite $\mathbf{X}_{\mathcal{I}}$. What's more, existing approaches to approximating the KL divergence via approximate gradients do not scale to high input or target dimensions (Sun et al., 2019).

4 Tractable and Scalable Function-Space Variational Inference in BNNs

The primary obstacle to making the objective in Equation (2) tractable is the KL divergence from $q_{f(\cdot;\boldsymbol{\Theta})}$ to $p_{f(\cdot;\boldsymbol{\Theta})}$. In this section, we describe how to obtain a tractable approximation to the variational objective.

4.1 Variational Distribution

We approach this problem by defining a suitable variational distribution over functions. Letting $\mathbf{X}_* \doteq \mathcal{X} \backslash \{\mathbf{X}_{\mathcal{D}}, \mathbf{X}_{\mathcal{I}}\}$ and $\mathbf{X}_{\mathcal{D}} \cap \mathbf{X}_{\mathcal{I}} = \varnothing$ so that $\mathbf{X}_*$ is an infinite collection of input points and $\mathbf{X}_{\mathcal{I}}$ is a finite set of *inducing input* points distinct from the training points, we define the variational distribution over functions as the stochastic process

$$q_{f(\mathbf{X}_*;\boldsymbol{\Theta}), f(\mathbf{X}_{\mathcal{D}};\boldsymbol{\Theta}), f(\mathbf{X}_{\mathcal{I}};\boldsymbol{\Theta})} \tag{4}$$

induced by a variational distribution over parameters $q_{\boldsymbol{\Theta}}$, which we define to be a mean-field Gaussian distribution: $q_{\boldsymbol{\Theta}} = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. To simplify notation, in the remainder of the paper, we denote the joint distribution over $f(\mathbf{X}_{\mathcal{D}}; \boldsymbol{\Theta})$ and $f(\mathbf{X}_{\mathcal{I}}; \boldsymbol{\Theta})$ more succinctly by $q_{f([\mathbf{X}_{\mathcal{D}}, \mathbf{X}_{\mathcal{I}}];\boldsymbol{\Theta})}$.

We follow Sun et al. (2019) and express the KL divergence in terms of the supremum over finite-dimensional marginals. Under the variational distribution defined above, we then get the objective function in Equation (2) simplifies to

$$\begin{aligned}\mathcal{F}(q_{\boldsymbol{\Theta}}, \mathbf{X}_{\mathcal{I}}, \mathbf{X}_{\mathcal{D}}, \mathbf{y}_{\mathcal{D}}) \doteq \mathbb{E}_{q_{f(\mathbf{X}_{\mathcal{D}};\boldsymbol{\Theta})}}[\log p(\mathbf{y}_{\mathcal{D}} \mid f(\mathbf{X}_{\mathcal{D}}; \boldsymbol{\theta}))] \\ - \sup_{n \in \mathbb{N}, \mathbf{X}_{\mathcal{I}} \in \mathcal{X}^n} \mathbb{D}_{\mathrm{KL}}(q_{f([\mathbf{X}_{\mathcal{D}}, \mathbf{X}_{\mathcal{I}}];\boldsymbol{\Theta})} \,\|\, p_{f([\mathbf{X}_{\mathcal{D}}, \mathbf{X}_{\mathcal{I}}];\boldsymbol{\Theta})}),\end{aligned} \tag{5}$$

where the KL divergence between distributions over functions is now reduced to a KL divergence between distributions over multivariate random variables, $q_{f([\mathbf{X}_{\mathcal{D}}, \mathbf{X}_{\mathcal{I}}];\boldsymbol{\Theta})}$ and $p_{f([\mathbf{X}_{\mathcal{D}}, \mathbf{X}_{\mathcal{I}}];\boldsymbol{\Theta})}$. Unlike the

variational objective considered in Sun et al. (2019), this variational objective is in fact a lower bound on the log-marginal likelihood $\log p(\mathbf{y}_\mathcal{D} \mid \mathbf{X}_\mathcal{D})$. For a full derivation of this result, see Appendix A.

4.2 Approximating the KL Divergence

Unfortunately, the KL divergence in Equation (5) is still intractable and cannot be estimated in a straightforward fashion, (i) since we can only sample from $q_{f([\mathbf{X}_\mathcal{D},\mathbf{X}_\mathcal{I}];\boldsymbol{\Theta})}$ and $p_{f([\mathbf{X}_\mathcal{D},\mathbf{X}_\mathcal{I}];\boldsymbol{\Theta})}$, but do not have access to their probability density functions, and (ii) since we cannot solve for the supremum.

We approach the problem of computing the KL divergence between two BNNs evaluated at a finite set of points first. To obtain an estimator of the intractable KL divergence $\mathbb{D}_{\mathrm{KL}}(q_{f([\mathbf{X}_\mathcal{D},\mathbf{X}_\mathcal{I}];\boldsymbol{\Theta})} \,\|\, p_{f([\mathbf{X}_\mathcal{D},\mathbf{X}_\mathcal{I}];\boldsymbol{\Theta})})$, we approximate $q_{f([\mathbf{X}_\mathcal{D},\mathbf{X}_\mathcal{I}];\boldsymbol{\Theta})}$ and $p_{f([\mathbf{X}_\mathcal{D},\mathbf{X}_\mathcal{I}];\boldsymbol{\Theta})}$ by computing the first-order Taylor expansion of the mapping f about the mean parameters of $q_{\boldsymbol{\Theta}}$ and $p_{\boldsymbol{\Theta}}$, respectively. To approximate the probability densities of $q_{f([\mathbf{X}_\mathcal{D},\mathbf{X}_\mathcal{I}];\boldsymbol{\Theta})}$ and $p_{f([\mathbf{X}_\mathcal{D},\mathbf{X}_\mathcal{I}];\boldsymbol{\Theta})}$, we use the following result:

Proposition 1 (Distribution under Linearized Mapping)**.** *For a stochastic function $f(\cdot\,;\boldsymbol{\Theta})$, stochastic parameters $\boldsymbol{\Theta}$ with mean $\mathbf{m} \doteq \mathbb{E}[\boldsymbol{\Theta}]$, and Jacobian $\mathcal{J}(\cdot, \mathbf{m}) \doteq \frac{\partial f(\cdot\,;\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}}|_{\boldsymbol{\Theta}=\mathbf{m}}$, denote the linearization of the stochastic function $f(\cdot\,;\boldsymbol{\Theta})$ about $\mathbf{m}$ by*

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

If $\boldsymbol{\Theta} \sim g_{\boldsymbol{\Theta}}$ and $g_{\boldsymbol{\Theta}}$ is a multivariate Gaussian distribution with mean $\mathbf{m}$ and diagonal co-variance $\mathbf{S}$, 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

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

and the distribution $\tilde{g}$ over $\tilde{f}(\mathbf{X}\,;\boldsymbol{\Theta})$ is given by

$$\tilde{g}_{\tilde{f}(\mathbf{X}\,;\boldsymbol{\Theta})} = \mathcal{N}(f(\mathbf{X}\,;\mathbf{m}), \mathcal{J}(\mathbf{X},\mathbf{m})\mathbf{S}\mathcal{J}(\mathbf{X}',\mathbf{m})^\top). \quad (6)$$

Proof. See Appendix A. □

Proposition 1 tells us 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 input points will be Gaussian as well. We use this insight to approximate the KL divergence in the variational objective by $\mathbb{D}_{\mathrm{KL}}(\tilde{q}_{\tilde{f}([\mathbf{X}_\mathcal{D},\mathbf{X}_\mathcal{I}];\boldsymbol{\Theta})} \,\|\, \tilde{p}_{\tilde{f}([\mathbf{X}_\mathcal{D},\mathbf{X}_\mathcal{I}];\boldsymbol{\Theta})})$, which is a KL divergence between two multivariate Gaussian distributions and as such analytically tractable. Under this approximation, we obtain the approximate variational objective

$$\tilde{\mathcal{F}}(q_{\boldsymbol{\Theta}}, \mathbf{X}_\mathcal{D}, \mathbf{y}_\mathcal{D}) \doteq \mathbb{E}_{q_{f(\mathbf{X}_\mathcal{D};\boldsymbol{\Theta})}}[\log p(\mathbf{y}_\mathcal{D} \mid f(\mathbf{X}_\mathcal{D};\boldsymbol{\theta}))] - \sup_{n\in\mathbb{N},\mathbf{X}_\mathcal{I}\in\mathcal{X}^n} \mathbb{D}_{\mathrm{KL}}(\tilde{q}_{\tilde{f}([\mathbf{X}_\mathcal{D},\mathbf{X}_\mathcal{I}];\boldsymbol{\Theta})} \,\|\, \tilde{p}_{\tilde{f}([\mathbf{X}_\mathcal{D},\mathbf{X}_\mathcal{I}];\boldsymbol{\Theta})}). \quad (7)$$

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 $p_{\boldsymbol{\Theta}}$.

To approximate the supremum, we propose a simple estimator. Specifically, for

$$I(\mathbf{X}_\mathcal{I}) \doteq \mathbb{D}_{\mathrm{KL}}(\tilde{q}_{\tilde{f}([\mathbf{X}_\mathcal{D},\mathbf{X}_\mathcal{I}];\boldsymbol{\Theta})} \,\|\, \tilde{p}_{\tilde{f}([\mathbf{X}_\mathcal{D},\mathbf{X}_\mathcal{I}];\boldsymbol{\Theta})})$$

we estimate the supremum from a finite sample by

$$\sup_{n\in\mathbb{N},\mathbf{X}_\mathcal{I}\in\mathcal{X}^n} I(\mathbf{X}_\mathcal{I}) \approx \log \sum_{\{\mathbf{X}_\mathcal{I}^{(i)}\}_{i=1}^S} \exp I(\mathbf{X}_\mathcal{I}^{(i)}) \quad (8)$$

with $\mathbf{X}_\mathcal{I}^{(i)} \sim p_{\mathbf{X}_\mathcal{I}}$. Since

$$\log \sum_{\{\mathbf{X}_\mathcal{I}^{(i)}\}_{i=1}^S} \exp I(\mathbf{X}_\mathcal{I}) \geq \max_{\{\mathbf{X}_\mathcal{I}^{(i)}\}_{i=1}^S} I(\mathbf{X}_\mathcal{I}^{(i)}),$$

this estimator is a biased estimator of the supremum even if $p_{\mathbf{X}_\mathcal{I}}$ would be guaranteed to draw uniform samples from $\mathcal{X}$. While the estimator is likely to only provide a very rough approximation, it encourages the variational distribution over functions to match the prior distribution over functions on sets of inducing inputs where the marginals differ the most. However, unlike a hard max-operator, the logsumexp estimator takes into account the contributions of the KL divergences for every $\mathbf{X}_\mathcal{I}^{(i)}$ in the sample. For details on how we choose $p_{\mathbf{X}_\mathcal{I}}$ in practice, see Appendix B.

4.3 Estimating the Variational Objective

Let $(\mathbf{X}_\mathcal{B}, \mathbf{y}_\mathcal{B})$ be a mini-batch of the training data and reparameterize $\boldsymbol{\Theta}$ as $\hat{\boldsymbol{\Theta}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\epsilon}^{(i)}) \doteq \boldsymbol{\mu} + \boldsymbol{\Sigma} \odot \boldsymbol{\epsilon}^{(i)}$. Using the estimator defined in Equation (8) and estimating the expected log-likelihood via Monte Carlo sampling, we then obtain the variational objective

$$\bar{\mathcal{F}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \{\mathbf{X}_\mathcal{I}\}_i^S, \mathbf{X}_\mathcal{D}, \mathbf{y}_\mathcal{D}) = \frac{1}{M}\sum_{j=1}^M \log p(\mathbf{y}_\mathcal{B} \mid f(\mathbf{X}_\mathcal{B}; \hat{\boldsymbol{\Theta}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\epsilon}^{(j)}))) - \log \sum_{\{\mathbf{X}_\mathcal{I}^{(i)}\}_{i=1}^S} \exp \mathbb{D}_{\mathrm{KL}}(\tilde{q}_{\tilde{f}([\mathbf{X}_\mathcal{B},\mathbf{X}_\mathcal{I}^{(i)}];\boldsymbol{\Theta})} \,\|\, \tilde{p}_{\tilde{f}([\mathbf{X}_\mathcal{B},\mathbf{X}_\mathcal{I}^{(i)}];\boldsymbol{\Theta})}) \quad (9)$$

with $\boldsymbol{\epsilon}^{(j)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_P)$ and $\mathbf{X}_\mathcal{I}^{(i)} \sim p_{\mathbf{X}_\mathcal{I}}$.

Table 1: Comparison of in- and out-of-distribution performance metrics (mean ± standard error over ten random seeds). Best results are printed in boldface. For further details about model architectures and training, see Appendix B. AUROC for binary in- and out-of-distribution detection on MNIST/NotMNIST.

Dataset	Method	Accuracy ↑	NLL ↓	ECE ↓	OOD-AUROC (M/NM) ↑
FMNIST	MAP	91.73±0.08	0.288±0.003	0.037±0.001	87.00±0.30 / 74.85±1.31
	MFVI	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)	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
	DUQ	92.40±0.20	–	–	95.50±0.70 / 94.60±1.80
	BNN-GLM	92.25±0.10	0.244±0.003	0.012±0.003	95.55±0.60 / –
	FSVI	**93.15**±0.13	**0.203**±0.004	0.014±0.002	**96.55**±0.41 / **95.27**±0.63
	FSVI (MAP init)	90.46±0.06	0.299±0.003	**0.009**±0.001	93.40±0.42 / 92.19±0.39
	SWAG	92.56±0.05	0.300±0.000	0.043±0.001	85.18±0.35 / 80.31±0.30
	Deep Ensemble	92.49±0.01	0.242±0.001	**0.019**±0.000	89.22±0.09 / 83.17±0.91
	MFVI Ensemble	92.46±0.04	0.294±0.001	0.026±0.000	94.29±0.21 / 90.31±0.37
	FSVI Ensemble	**94.44**±0.07	**0.181**±0.001	0.020±0.001	**97.85**±0.15 / **96.95**±0.20

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

$$q(\mathbf{y}_* \mid \mathbf{x}_*) = \int p(\mathbf{y}_* \mid f(\mathbf{x}_*; \boldsymbol{\theta}))\, q_{f(\mathbf{x}_*;\boldsymbol{\Theta})}\, df(\mathbf{x}_*; \boldsymbol{\Theta}) \approx \frac{1}{M} \sum_{j=1}^{M} p(\mathbf{y}_* \mid f(\mathbf{x}_*; \boldsymbol{\Theta}^{(j)})), \boldsymbol{\Theta}^{(j)} \sim q_{\boldsymbol{\Theta}}. \tag{10}$$

5 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; Burt et al., 2021; Jacot et al., 2018; Pan et al., 2020; Sun et al., 2019; Titsias et al., 2020).

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 at all (Wang et al., 2019). More recent work has attempted to circumvent the intractability of the variational objective in Equation (2) by proposing alternative objectives derived from the weight space-function space duality in BNNs and Bayesian linear models (Ma et al., 2019; Ober and Aitchison, 2020). In a similar vein, Immer et al. (2020) follow Khan et al. (2019) to explore the distribution over functions induced by different BNN models and show that approximating BNN posterior distribution via the Laplace and Generalized-Gauss-Newton approximation corresponds to a certain type of linearization. Crucially, unlike in our approach, they use the linearization to change the model, whereas we only use a linearization to obtain a tractable estimator for one term in the variational objective. Furthermore, they do not consider a linearization about the mean of a variational distribution but instead linearize about the maximum a posteriori (MAP) parameters obtained by training a deterministic neural network with Tikhonov regularization.

Burt et al. (2021) consider the function-space variational objective in Equation (2), and highlight advantages and limitations of employing the KL divergence in this setting. They show that minimizing the KL divergence between a wide class of parametric distributions, such as those induced by a finite-width BNN, and the posterior induced by a (non-degenerate) BNN can result in an ill-defined objective. A complementary 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).

6 Empirical Evaluation

We evaluate FSVI on a wide array of high-dimensional classification tasks prior work on function-space variational inference (Sun et al., 2019) was unable to scale to. 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 comprehensive description of the experiment setups, details on the models, training, and validation procedures, see Appendix B. We provide additional visualizations for a larger set of datasets and a larger selection of models in Appendix D.

6.1 FashionMNIST & CIFAR-10

In this set of experiments, we assess the reliability of the uncertainty estimates generated by FSVI. If a BNN

Table 2: Comparison of in- and out-of-distribution performance metrics (mean ± standard error over ten random seeds). Best results are printed in boldface. For further details about model architectures and training, see Appendix B. AUROC for binary in- and out-of-distribution detection on SVHN.

Dataset	Method	Accuracy ↑	NLL ↓	ECE ↓	OOD-AUROC/C-CIFAR Accuracy ↑
			CNN		
CIFAR-10	MAP	87.35±0.09	0.491±0.005	0.070±0.001	90.86±0.43 / 74.20±0.60
	MFVI	84.04±0.07	0.372±0.002	**0.016**±0.001	92.62±0.31 / 71.48±0.74
	MFVI (tempered)	86.29±0.08	0.457±0.003	0.049±0.001	91.54±0.57 / 72.02±0.58
	MFVI (radial)	83.99±0.18	0.510±0.001	0.048±0.002	86.04±0.18 / 73.54±0.53
	MC DROPOUT	83.89±0.18	0.412±0.005	0.018±0.002	92.69±0.49 / 69.75±0.82
	BNN-GLM	81.37±0.15	0.601±0.008	0.084±0.010	84.30±0.02 / –
	FSVI	86.34±0.11	0.499±0.005	0.061±0.001	94.00±0.39 / 73.59±0.66
	FSVI (MAP init)	**88.10**±0.08	**0.330**±0.003	0.021±0.001	**97.71**±0.11 / **79.64**±0.36
	SWAG	89.73±0.14	0.480±0.001	0.067±0.002	89.79±0.50 / 76.12±0.51
	Deep Ensemble	89.28±0.04	0.339±0.003	0.020±0.001	92.00±0.16 / 76.65±0.21
	MFVI Ensemble	89.49±0.07	0.330±0.001	0.049±0.001	94.06±0.20 / 73.67±0.38
	FSVI Ensemble	**90.17**±0.03	**0.314**±0.001	**0.018**±0.001	**96.17**±0.10 / **78.63**±0.40
			ResNet-18		
	MAP	92.19±0.15	0.307±0.006	0.046±0.001	95.17±0.40 / 78.55±1.01
	MFVI	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	0.31±0.004	0.041±0.001	90.32±0.57 / 80.19±1.44
	DUQ	94.10±0.2	–	–	92.70±1.30 / –
	VOGN	84.27±0.20	0.477±0.006	0.040±0.002	87.60±0.20 / –
	FSVI	93.30±0.04	0.295±0.003	0.034±0.001	94.89±0.19 / 80.88±0.47
	FSVI (MAP init)	**94.10**±0.04	**0.175**±0.002	**0.014**±0.001	**98.89**±0.23 / **81.38**±0.41
	Deep Ensemble	95.13±0.06	0.158±0.001	0.019±0.001	98.04±0.07 / 81.22±0.37
	FSVI Ensemble	**95.29**±0.03	**0.150**±0.003	**0.011**±0.001	**99.12**±0.36 / **81.44**±0.43

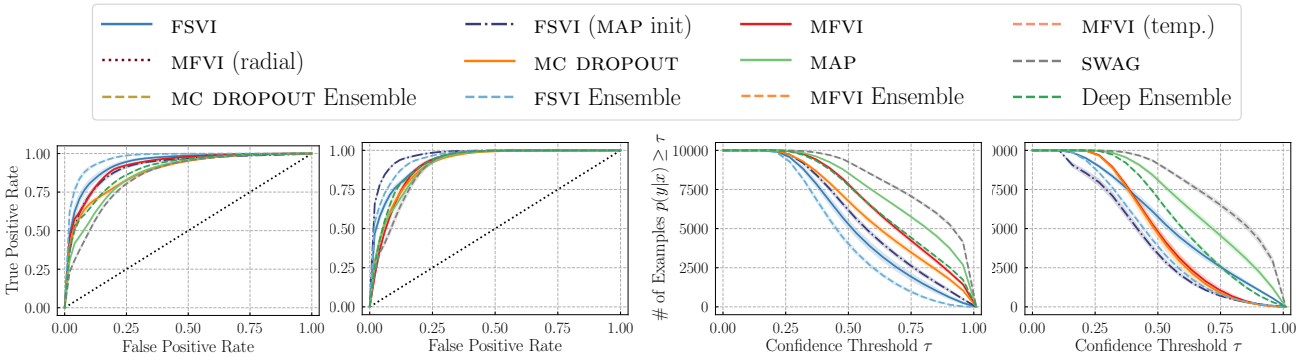
trained via FSVI is able to perform reliable uncertainty estimation, its predictions will be significantly higher on input points that were generated according to a different data-generating distribution than the training data. For models trained on the FashionMNIST dataset, we use the MNIST and NotMNIST 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 and plot it along with the out-of-distribution predictive confidence of different methods in Figure 2.

Choice of prior. For all experiments that involve uncertainty quantification, we chose a prior distribution over parameters that would induce a prior distribution over functions $p_{f(\cdot;\boldsymbol{\Theta})}$ that has high uncertainty on input points far away from the training points. For further details, see Appendices B and F.

Predictive Performance & Calibration. To assess in-distribution predictive performance and calibration, we report test accuracies, test negative log-likelihoods, and test expected calibration errors for models trained on FashionMNIST and CIFAR-10 in Tables 1 and 2. As can be seen from the results, FSVI outperforms other non-ensemble methods in terms of test accuracy and test negative log-likelihood on both datasets and even outperforms other ensemble methods on FashionMNIST. Across all both datasets and all models, FSVI ensembles achieve the best performance. We observe, however, that FSVI does not consistently lead to well-calibrated predictive distributions. On both datasets, vanilla FSVI results in good but not outstanding calibration.

Out-of-Sample Uncertainty Estimation. In Tables 1 and 2 and Figure 2, we report the results from our assessment of different models' predictive uncertainty estimates. FSVI stands out among all methods, including various ensemble methods for its ability to generate reliable predictive uncertainty estimates that allow distinguishing between in- and out-of-distribution inputs. Furthermore, Figure 2 ((c) and



(a) ROC (FMNIST) **(b)** ROC (CIFAR-10) **(c)** Confidence (FMNIST) **(d)** Confidence (CIFAR-10)

Figure 2: Uncertainty evaluation metrics for out-of-distribution predictions. Models were trained on either Fashion-MNIST or CIFAR-10, and MNIST and SVHN are used as out-of-distribution data. **(a) and (b)**: Receiver operating characteristic curves for out-of-distribution detection. Curves closer to the top left are better. **(c) and (d)**: Model confidence on out-of-distribution inputs. Curves closer to the bottom left are better.

(d)) show that FSVI is also more likely to generate low-confidence predictions on out-of-distribution inputs compared to related methods.

6.2 Rotated MNIST & Corrupted CIFAR-10

In order for a model to be considered reliable, we would like it (i) to exhibit low predictive uncertainty on training data and high predictive uncertainty on out-of-distribution inputs, (ii) to have the ability to use its predictive uncertainty estimates to distinguish in- from out-of-distribution data, and (iii) if possible, maintain high predictive accuracy even under distribution shift (Ovadia et al., 2019). A model that satisfies all of these desiderata would be able to alert a human in the loop or be referred to a domain expert when it encounters data points on which it has particularly high predictive uncertainty. This ability increases a model's level of robustness to making poor but confident predictions and is particularly relevant in safety-critical settings, such as medical diagnosis.

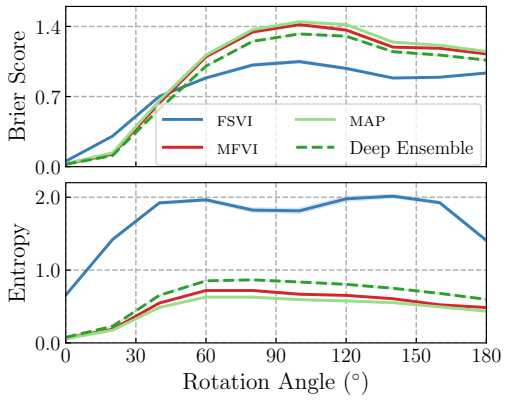


Figure 3: Predictive uncertainty and accuracy on rotated MNIST. Models with reliable uncertainty estimates would exhibit higher predictive uncertainty the more the digits are rotated. Ideally, such models would maintain high predictive accuracy (low Brier score).

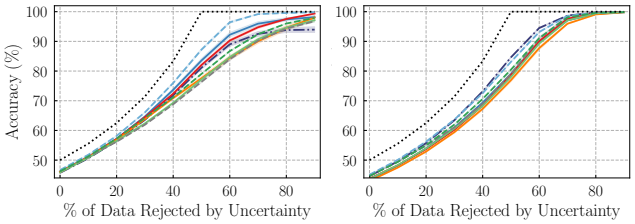
To illustrate these desiderata, we consider the rotated MNIST task (Ovadia et al., 2019) where the goal is to maintain a high level of predictive accuracy (measured in terms of Brier scores, which is more sensitive to poorly calibrated predictions than a model's accuracy) while exhibiting high predictive uncertainty. Figure 3 shows Brier scores (lower is better) and predictive entropy (higher means more uncertain) of four different models. Rotating the MNIST digits gradually shifts the data distributions, we would expect Brier scores to increase (worse predictive accuracy) on the increasingly rotated digits. A good 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 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 is most robust to distribution shift and achieves the highest accuracy on the corrupted data.

6.3 Downstream Task: Selective Prediction

In some safety-critical real-world applications of machine learning systems, such as medical diagnosis, it is possible to refer a subset of data points for which a model is supposed to make a prediction to an expert for review. In such settings, reliable uncertainty estimation can be used to identify data points at which a

model is most uncertain and refer them to an expert and only make predictions on data points for which the model has a high degree of certainty. On such selective prediction tasks, a model's predictive accuracy and uncertainty estimates are required to work in tandem to select samples and make correct predictions on them.

We follow van Amersfoort et al. (2020) and simulate this setting by adding evaluation points taken from the MNIST dataset to the FashionMNIST test dataset and by adding evaluation points taken from the SVHN dataset to the CIFAR-10 test dataset. The goal for this task is then to achieve as high a predictive accuracy as possible on the evaluation points not rejected based on the model's uncertainty. We plot the results in Figure 4, where we included a dashed line to indicate the performance of an optimal classifier. As can be seen in the plot, FSVI (the blue lines) outperforms related methods for both dataset pairs.



(a) FMNIST/MNIST **(b)** CIFAR-10/SVHN

Figure 4: Selective prediction accuracy. Curves closer to the theoretical maximum (denoted by the dotted line) are better. The legend can be found in Figure 2.

6.4 Downstream Task: Continual Learning

Continual learning is the process of developing new abilities while retaining existing ones. Sequential Bayesian inference over predictive functions is a natural framework for doing this, but applying it to deep neural networks is challenging in practice. To prevent forgetting previously learned predictive abilities over a sequence of tasks $t = 1, 2, 3...$, we set the prior distribution over functions to the variational distribution over functions induced by the distribution over parameters learned on the previous task, that is,

$$\begin{aligned}&\bar{\mathcal{F}}(\boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)}, \{\mathbf{X}_\mathcal{I}\}_i^S, \mathbf{X}_\mathcal{D}, \mathbf{y}_\mathcal{D}) \\ &= \frac{1}{M}\sum_{j=1}^{M} \log p(\mathbf{y}_\mathcal{B}^{(t)} \mid f(\mathbf{X}_\mathcal{B}^{(t)}; \hat{\boldsymbol{\Theta}}(\boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)}, \boldsymbol{\epsilon}^{(j)}))) \quad (11) \\ &- \log \sum_{\{\mathbf{X}_\mathcal{I}^{(i)}\}_{i=1}^S} \exp \mathbb{D}_{\mathrm{KL}}(\tilde{q}^{(t)}_{\tilde{f}([\mathbf{X}_\mathcal{D}^{(t)}, \mathbf{X}_\mathcal{I}^{(i)}]; \boldsymbol{\Theta})} \,\|\, \tilde{q}^{(t-1)}_{\tilde{f}([\mathbf{X}_\mathcal{D}^{(t)}, \mathbf{X}_\mathcal{I}^{(i)}]; \boldsymbol{\Theta})})\end{aligned}$$

with $\boldsymbol{\epsilon}^{(j)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_P)$ and $\mathbf{X}_\mathcal{I}^{(i)} \sim p_{\mathbf{X}_\mathcal{I}}$.

We follow Farquhar et al. (2020a) and compare our method to a widely used parameter-space method for continual learning, variational continual learning (VCL; (Nguyen et al., 2018)), used with MFVI and radial MFVI. As can be seen in Figure 5, FSVI performs significantly better at incorporating prior information learned on previous tasks as evidenced by the consistently high accuracy even on tasks learned in the past (here: tasks 1–4). These results demonstrate the extent to which BNNs trained via FSVI benefit from the ability to incorporate meaningful prior information about the underlying task. Further details about the experiment setup can be found in Appendix B.

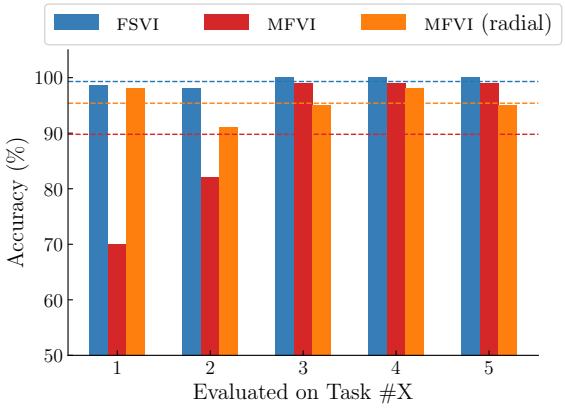


Figure 5: Predictive accuracy on five different tasks. Models are trained on five FMNIST tasks in sequence using the posterior distribution over functions from the previous task for FSVI and the posterior distribution over parameters from the previous task for MFVI. Both parameter-space inference methods use variational continual learning (VCL; (Nguyen et al., 2018)).

7 Conclusion

We proposed a new approach to variational inference in BNNs, where the parameters are inferred *indirectly* by performing inference over an induced distribution over functions. The proposed variational objective is easy to compute, and we demonstrated that it can be scaled up to high-dimensional data and large neural networks. We further showed that FSVI exhibits competitive in- and out-of-distribution predictive performance on a wide range of datasets when compared to well-established and state-of-the-art methods. To demonstrate the versatility of the proposed approach and to showcase the usefulness of incorporating empirical prior distributions over functions, we evaluated FSVI on two downstream tasks and showed that it outperforms state-of-the-art Bayesian and non-Bayesian methods. We hope that this work will lead to further research into function-space variational inference and real-world applications of it in areas such as active learning or federated learning.

References

Benjamin, A., Rolnick, D., and Kording, K. (2019). Measuring and regularizing networks in function space. In *International Conference on Learning Representations*.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural network. volume 37 of *Proceedings of Machine Learning Research*, pages 1613–1622, Lille, France. PMLR.

Burt, D. R., Ober, S. W., Garriga-Alonso, A., and van der Wilk, M. (2021). Understanding variational inference in function-space. In *Third Symposium on Advances in Approximate Bayesian Inference*.

Farquhar, S., Osborne, M. A., and Gal, Y. (2020a). Radial bayesian neural networks: Beyond discrete support in large-scale bayesian deep learning. In Chiappa, S. and Calandra, R., 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.

Farquhar, S., Smith, L., and Gal, Y. (2020b). Liberty or depth: Deep bayesian neural nets do not need complex weight posterior approximations. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Foong, A. Y. K., Burt, D. R., Li, Y., and Turner, R. E. (2020). On the expressiveness of approximate inference in bayesian neural networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Foong, A. Y. K., Li, Y., Hernández-Lobato, J. M., and Turner, R. E. (2019). 'in-between' uncertainty in bayesian neural networks.

Graves, A. (2011). 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. Curran Associates Inc.

Hinton, G. E. and van Camp, D. (1993). 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. Association for Computing Machinery.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347.

Immer, A., Korzepa, M., and Bauer, M. (2020). Improving predictions of bayesian neural networks via local linearization.

Izmailov, P., Maddox, W. J., Kirichenko, P., Garipov, T., Vetrov, D., and Wilson, A. G. (2020). Subspace inference for bayesian deep learning. In Adams, R. P. and Gogate, V., editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 1169–1179. PMLR.

Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 8571–8580. Curran Associates, Inc.

Khan, M. E. E., Immer, A., Abedi, E., and Korzepa, M. (2019). Approximate inference turns deep networks into gaussian processes. In Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 3094–3104. Curran Associates, Inc.

Ma, C., Li, Y., and Hernandez-Lobato, J. M. (2019). Variational implicit processes. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4222–4233. PMLR.

MacKay, D. J. C. (1992). A practical bayesian framework for backpropagation networks. *Neural Comput.*, 4(3):448–472.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. (2019). A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, pages 13153–13164.

Matthews, A. G. d. G., Hensman, J., Turner, R., and Ghahramani, Z. (2016). 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. PMLR.

Neal, R. M. (1996). Bayesian Learning for Neural Networks.

Nguyen, C. V., Li, Y., Bui, T. D., and Turner, R. E. (2018). Variational continual learning. *International Conference on Learning Representations*.

Ober, S. W. and Aitchison, L. (2020). Global inducing point variational posteriors for bayesian neural networks and deep gaussian processes.

Osawa, K., Swaroop, S., Khan, M. E. E., Jain, A., Eschenhagen, R., Turner, R. E., and Yokota, R. (2019). Practical deep learning with bayesian principles. In Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32, pages 4287–4299. Curran Associates, Inc.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. (2019). Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems 32*.

Pan, P., Swaroop, S., Immer, A., Eschenhagen, R., Turner, R. E., and Khan, M. E. (2020). Continual deep learning by functional regularisation of memorable past. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Polyanskiy, Y. and Wu, Y. (2017). Strong data-processing inequalities for channels and bayesian networks. In Carlen, E., Madiman, M., and Werner, E. M., editors, *Convexity and Concentration*, pages 211–249, New York, NY. Springer New York.

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

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

Sun, S., Zhang, G., Shi, J., and Grosse, R. B. (2019). Functional variational bayesian neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Titsias, M. K., Schwarz, J., de G. Matthews, A. G., Pascanu, R., and Teh, Y. W. (2020). Functional regularisation for continual learning with gaussian processes. In *International Conference on Learning Representations*.

van Amersfoort, J., Smith, L., Jesson, A., Key, O., and Gal, Y. (2021). Variational deterministic uncertainty quantification.

van Amersfoort, J., Smith, L., Teh, Y. W., and Gal, Y. (2020). Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning*.

Wainwright, M. J. and Jordan, M. I. (2008). *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc., Hanover, MA, USA.

Wang, Z., Ren, T., Zhu, J., and Zhang, B. (2019). Function space particle optimization for bayesian neural networks. In *International Conference on Learning Representations*.

Wenzel, F., Roth, K., Veeling, B., Swiatkowski, J., Tran, L., Mandt, S., Snoek, J., Salimans, T., Jenatton, R., and Nowozin, S. (2020). How good is the Bayes posterior in deep neural networks really? In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10248–10259. PMLR.

Supplementary Material

Table of Contents

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

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

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

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

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

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

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

$$p(\mathbf{y}_\mathcal{D}) \geq \mathbb{E}_{Q_{\mathbf{X}_\mathcal{D}}}\left[\log p\left(\mathbf{y}_\mathcal{D} \mid f_{\mathbf{X}_\mathcal{D}}\right)\right] - \int \log \frac{dQ^\pi}{dP^\pi}(f) dQ^\pi(f) \tag{A.10}$$

$$= \mathbb{E}_{Q_{\mathbf{X}_\mathcal{D}}}\left[\log p\left(\mathbf{y}_\mathcal{D} \mid f_{\mathbf{X}_\mathcal{D}}\right)\right] - \mathbb{D}_{\mathrm{KL}}(Q_{\mathbf{X}_\mathcal{D}, \mathbf{X}_{\setminus\mathcal{D}}} \,\|\, P_{\mathbf{X}_\mathcal{D}, \mathbf{X}_{\setminus\mathcal{D}}}), \tag{A.11}$$

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

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

which corresponds to the expression for the function-space variational objective in Section 3.

A.2 Distribution under Linearized Function Mapping

Proposition 1 (Distribution under Linearized Mapping)**.** *For a stochastic function $f(\cdot\,; \boldsymbol{\Theta})$, stochastic parameters $\boldsymbol{\Theta}$ with mean $\mathbf{m} \doteq \mathbb{E}[\boldsymbol{\Theta}]$, and Jacobian $\mathcal{J}(\cdot, \mathbf{m}) \doteq \frac{\partial f(\cdot\,; \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}}|_{\boldsymbol{\Theta}=\mathbf{m}}$, denote the linearization of the stochastic function $f(\cdot\,; \boldsymbol{\Theta})$ about $\mathbf{m}$ by*

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

If $\boldsymbol{\Theta} \sim g_{\boldsymbol{\Theta}}$ and $g_{\boldsymbol{\Theta}}$ is a multivariate Gaussian distribution with mean $\mathbf{m}$ as defined above and diagonal co-variance $\mathbf{S}$, 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

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

and the distribution $\tilde{g}$ over $\tilde{f}(\mathbf{X}\,; \boldsymbol{\Theta})$ is given by

$$\tilde{g}_{\tilde{f}(\mathbf{X}\,; \boldsymbol{\Theta})} = \mathcal{N}(f(\mathbf{X}\,; \mathbf{m}), \mathcal{J}(\mathbf{X}, \mathbf{m}) \mathbf{S} \mathcal{J}(\mathbf{X}', \mathbf{m})^\top). \tag{A.13}$$

Proof. Since $\boldsymbol{\Theta} \sim \mathcal{N}(\mathbf{m}, \mathbf{S})$, and $\tilde{f}(\mathbf{X}; \boldsymbol{\Theta}) = f(\mathbf{X}; \mathbf{m}) + \mathcal{J}(\mathbf{X}; \mathbf{m})(\boldsymbol{\Theta} - \mathbf{m})$ is a linear transformation of $\boldsymbol{\Theta}$, $\tilde{f}(\mathbf{X}; \boldsymbol{\Theta})$ is a multivariate Gaussian distribution,

$$\tilde{g}(\tilde{f}(\mathbf{X}; \boldsymbol{\theta})) = \mathcal{N}(\widetilde{m}(\mathbf{X}), \widetilde{\mathrm{Cov}}(\mathbf{X}, \mathbf{X}')), \tag{A.14}$$

with some (as of yet unknown) predictive mean $\widetilde{m}(\mathbf{X})$ and predictive covariance $\widetilde{\mathrm{Cov}}(\mathbf{X}, \mathbf{X}')$. To find $\tilde{g}_{\tilde{f}(\mathbf{X}; \boldsymbol{\Theta})}$, we need to find the predictive mean $\widetilde{m}(\mathbf{X})$ and the predictive covariance $\widetilde{\mathrm{Cov}}(\mathbf{X}, \mathbf{X}')$, which, by definition, we can write as:

$$\widetilde{m}(\mathbf{X}) = \mathbb{E}[\tilde{f}(\mathbf{X}; \boldsymbol{\Theta})] \tag{A.15}$$

and

$$\widetilde{\mathrm{Cov}}(\mathbf{X}, \mathbf{X}') = \mathrm{Cov}(\tilde{f}(\mathbf{X}; \boldsymbol{\Theta}), \tilde{f}(\mathbf{X}'; \boldsymbol{\theta})) = \mathbb{E}[(\tilde{f}(\mathbf{X}; \boldsymbol{\theta}) - \mathbb{E}[\tilde{f}(\mathbf{X}; \boldsymbol{\theta})])\,(\tilde{f}(\mathbf{X}'; \boldsymbol{\theta}) - \mathbb{E}[\tilde{f}(\mathbf{X}'; \boldsymbol{\theta})])^\top]. \tag{A.16}$$

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

$$\widetilde{m}(\mathbf{X}) = \mathbb{E}[\tilde{f}(\mathbf{X}; \boldsymbol{\theta})] \tag{A.17}$$

$$= \mathbb{E}[f(\mathbf{X}; \mathbf{m}) + \mathcal{J}(\mathbf{X}; \mathbf{m})(\boldsymbol{\Theta} - \mathbf{m})] \tag{A.18}$$

$$= f(\mathbf{X}; \mathbf{m}) + \mathcal{J}(\mathbf{X}; \mathbf{m})(\mathbb{E}[\boldsymbol{\Theta}] - \mathbf{m}) \tag{A.19}$$

$$= f(\mathbf{X}; \mathbf{m}). \tag{A.20}$$

To see that $\widetilde{\mathrm{Cov}}(\mathbf{X}, \mathbf{X}') = \mathrm{Cov}(\tilde{f}(\mathbf{X}; \boldsymbol{\theta}), \tilde{f}(\mathbf{X}'; \boldsymbol{\theta})) = \mathcal{J}(\mathbf{X}; \mathbf{m}) \mathbf{S} \mathcal{J}(\mathbf{X}'; \mathbf{m})^\top$, 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,

$$\mathrm{Cov}(\tilde{f}(\mathbf{X}; \boldsymbol{\Theta}), \tilde{f}(\mathbf{X}'; \boldsymbol{\Theta})) = \mathbb{E}[\tilde{f}(\mathbf{X}; \boldsymbol{\Theta}) \tilde{f}(\mathbf{X}'; \boldsymbol{\Theta})^\top] - \mathbb{E}[\tilde{f}(\mathbf{X}; \boldsymbol{\Theta})]\mathbb{E}[\tilde{f}(\mathbf{X}'; \boldsymbol{\Theta})]^\top. \tag{A.21}$$

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

$$
\begin{aligned}
&\mathbb{E}_{g_{\boldsymbol{\Theta}}}[\tilde{f}(\mathbf{X};\boldsymbol{\Theta})\tilde{f}(\mathbf{X}';\boldsymbol{\Theta})^\top] \\
&\quad= \mathbb{E}_{g_{\boldsymbol{\Theta}}}[(f(\mathbf{X};\mathbf{m}) + \mathcal{J}(\mathbf{X};\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m}))(f(\mathbf{X}';\mathbf{m}) + \mathcal{J}(\mathbf{X}';\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m}))^\top] && \text{(A.22)} \\
&\quad= \mathbb{E}_{g_{\boldsymbol{\Theta}}}[f(\mathbf{X};\mathbf{m})f(\mathbf{X}';\mathbf{m})^\top + (\mathcal{J}(\mathbf{X};\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m}))(\mathcal{J}(\mathbf{X}';\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m}))^\top \\
&\qquad\qquad + f(\mathbf{X};\mathbf{m})(\mathcal{J}(\mathbf{X}';\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m}))^\top + \mathcal{J}(\mathbf{X};\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m})f(\mathbf{X}';\mathbf{m})^\top] && \text{(A.23)} \\
&\quad= \mathbb{E}_{g_{\boldsymbol{\Theta}}}[f(\mathbf{X};\mathbf{m})f(\mathbf{X}';\mathbf{m})^\top + \mathcal{J}(\mathbf{X};\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m})^\top\mathcal{J}(\mathbf{X}';\mathbf{m})^\top \\
&\qquad\qquad + f(\mathbf{X};\mathbf{m})(\mathcal{J}(\mathbf{X}';\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m}))^\top + \mathcal{J}(\mathbf{X};\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m})f(\mathbf{X}';\mathbf{m})^\top] && \text{(A.24)} \\
&\quad= f(\mathbf{X};\mathbf{m})f(\mathbf{X}';\mathbf{m})^\top + \mathcal{J}(\mathbf{X};\mathbf{m})\mathbb{E}_{g_{\boldsymbol{\Theta}}}[(\boldsymbol{\Theta}-\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m})^\top]\mathcal{J}(\mathbf{X}';\mathbf{m})^\top \\
&\qquad\qquad + f(\mathbf{X};\mathbf{m})(\mathcal{J}(\mathbf{X}';\mathbf{m})(\underbrace{\mathbb{E}_{g_{\boldsymbol{\Theta}}}[\boldsymbol{\Theta}]-\mathbf{m}}_{=0}))^\top + \mathcal{J}(\mathbf{X};\mathbf{m})(\underbrace{\mathbb{E}_{g_{\boldsymbol{\Theta}}}[\boldsymbol{\Theta}]-\mathbf{m}}_{=0})f(\mathbf{X}';\mathbf{m})^\top, && \text{(A.25)}
\end{aligned}
$$

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

$$
\begin{aligned}
\mathbb{E}_{g_{\boldsymbol{\Theta}}}[\tilde{f}(\mathbf{X};\boldsymbol{\Theta})\tilde{f}(\mathbf{X}';\boldsymbol{\Theta})^\top] &= f(\mathbf{X};\mathbf{m})f(\mathbf{X}';\mathbf{m})^\top + \mathcal{J}(\mathbf{X};\mathbf{m})\mathbb{E}_{g_{\boldsymbol{\Theta}}}[(\boldsymbol{\Theta}-\mathbf{m})(\boldsymbol{\Theta}-\mathbf{m})^\top]\mathcal{J}(\mathbf{X}';\mathbf{m})^\top && \text{(A.26)} \\
&= f(\mathbf{X};\mathbf{m})f(\mathbf{X};\mathbf{m})^\top + \mathcal{J}(\mathbf{X};\mathbf{m})\,\mathbb{V}[\boldsymbol{\Theta}]\mathcal{J}(\mathbf{X}';\mathbf{m})^\top. && \text{(A.27)}
\end{aligned}
$$

With this result, we obtain the covariance function

$$
\begin{aligned}
\widetilde{\mathrm{Cov}}(\mathbf{X},\mathbf{X}') &= \mathrm{Cov}(\tilde{f}(\mathbf{X};\boldsymbol{\Theta}),\tilde{f}(\mathbf{X}';\boldsymbol{\Theta})) && \text{(A.28)} \\
&= \mathbb{E}[\tilde{f}(\mathbf{X};\boldsymbol{\Theta})\tilde{f}(\mathbf{X}';\boldsymbol{\Theta})^\top] - \mathbb{E}[\tilde{f}(\mathbf{X};\boldsymbol{\Theta})]\mathbb{E}[\tilde{f}(\mathbf{X}';\boldsymbol{\Theta})]^\top && \text{(A.29)} \\
&= \mathbb{E}[\tilde{f}(\mathbf{X};\boldsymbol{\Theta})\tilde{f}(\mathbf{X}';\boldsymbol{\Theta})^\top] - f(\mathbf{X};\mathbf{m})f(\mathbf{X};\mathbf{m})^\top + \mathcal{J}(\mathbf{X};\mathbf{m})\,\mathbb{V}[\boldsymbol{\Theta}]\mathcal{J}(\mathbf{X}';\mathbf{m})^\top && \text{(A.30)} \\
&= f(\mathbf{X};\boldsymbol{\Theta})f(\mathbf{X}';\boldsymbol{\Theta})^\top - f(\mathbf{X};\mathbf{m})f(\mathbf{X};\mathbf{m})^\top + \mathcal{J}(\mathbf{X};\mathbf{m})\,\mathbb{V}[\boldsymbol{\Theta}]\mathcal{J}(\mathbf{X}';\mathbf{m})^\top && \text{(A.31)} \\
&= \mathcal{J}(\mathbf{X};\mathbf{m})\,\mathbb{V}[\boldsymbol{\Theta}]\mathcal{J}(\mathbf{X}';\mathbf{m})^\top. && \text{(A.32)}
\end{aligned}
$$

Finally, $\mathbb{V}[\boldsymbol{\Theta}] = \mathbf{S}$ yields

$$
\widetilde{\mathrm{Cov}}(\mathbf{X},\mathbf{X}') = \mathcal{J}(\mathbf{X};\mathbf{m})\mathbf{S}\mathcal{J}(\mathbf{X}';\mathbf{m})^\top, \tag{A.33}
$$

where $\mathbf{S}$ is a diagonal matrix. This concludes the proof. □

Appendix B Model, Algorithmic & Experimental Details

B.1 Hyperparameter Selection

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 inducing inputs used to evaluate the KL divergence, the inducing input sampling methods, and the number of Monte Carlo samples used to evaluate the expected log-likelihood. The full results can be found in Appendix F. 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 inducing inputs 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 in Tables 3, 4, and 5.

B.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, we used a prior variance of $\mathbf{\Sigma}_0 = 10$ and sampled 50% of the inducing inputs for each gradient step from the mini-batch and the other 50% according to the method described in Appendix B.8.

B.3 CIFAR-10/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 network architecture with six convolutional layers of 32, 32, 64, 64, 128, 128 3×3 filters and a fully-connected final layer of 128 hidden units is used. A max pooling operation is placed after the second, fourth, and sixth convolutional layer and ReLU activations are 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 200 and a learning rate 5×10^{-4}.

For FSVI, we used a prior variance of $\mathbf{\Sigma}_0 = 0.1$ and sampled the inducing inputs for each gradient step according to the method described in Appendix B.8.

B.4 Continual Learning

Split FashionMNIST consists of five tasks, where each task is binary classification on a pair of MNIST classes. 60,000 data samples are used for training and 10,000 data samples are used for testing. The input images are converted to floating-point numbers with values in the range $[0, 1]$. To ensure fair comparison to VCL (with standard and radial MFVI), we use the same neural-network size across methods. We use fully connected neural networks, with four hidden layers of size 200. ReLU activation functions are applied to non-output units. We use a single-head setup and do not use a coreset. For FSVI, when training on each task, 40 inducing inputs are chosen uniformly randomly from the training data of the current task.

For the first task, FSVI uses a prior distribution over functions with fixed mean and diagonal covariance. The prior distribution is assumed to be Gaussian with zero mean and a diagonal covariance of magnitude 100. The number of epochs on each task is 60, 60, 10, 80, respectively. The batch size is 128. The predictive distribution used for computing the expected log-likelihood is estimated using five Monte Carlo samples.

B.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 inducing input randomly from $[-10, 10]^2$.

B.6 1D Regression Problems

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 inducing input randomly from $[-10, 10]$.

B.7 Further Implementation Details

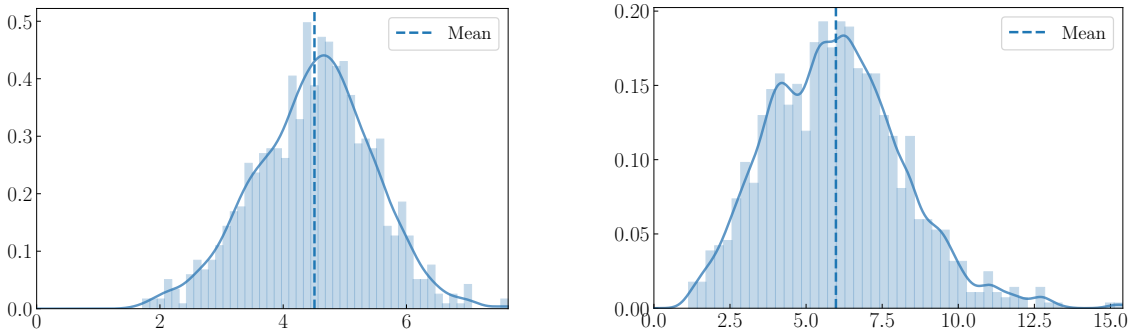
We use the Adam optimizer with default settings of $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\epsilon = 10^{-8}$ for all experiments. The deterministic neural networks that were used for the ensemble were trained with a weight decay of $\lambda =$ 1e-1. MFVI (tempered) was trained with a KL scaling factor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

B.8 Selection of Inducing Inputs.

We estimate the supremum at every gradient step by sampling a set of inducing inputs $\mathbf{X}_\mathcal{I}$ from a distribution $p_{\mathbf{X}_\mathcal{I}}$ at every gradient step. For tasks with image inputs, $p_{\mathbf{X}_\mathcal{I}}$ is defined as a uniform distribution over images with monochromatic channels with the color of each channel distributed according to the empirical pixel value distribution of each channel for images in the training data. For regression tasks with a D-dimensional input space, $p_{\mathbf{X}_\mathcal{I}}$ 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 D.

Appendix C Validation of Approximations

C.1 Validation of Linearization Assumption: FashionMNIST



(a) $\mathbb{D}_{\mathrm{KL}}(q(f(\mathbf{X}_{\mathcal{E}};\boldsymbol{\theta}))\,\|\,\tilde{q}(\tilde{f}(\mathbf{X}_{\mathcal{E}};\boldsymbol{\theta})))$ **(b)** $\mathbb{D}_{\mathrm{KL}}(\tilde{q}(\tilde{f}(\mathbf{X}_{\mathcal{E}};\boldsymbol{\theta}))\,\|\,q(f(\mathbf{X}_{\mathcal{E}};\boldsymbol{\theta})))$

Figure 6: KL divergences between distributions induced by $q_{\boldsymbol{\Theta}}$ under linearized and non-linearized mappings evaluated on 1,000 data points $\mathbf{X}_{\mathcal{E}}$ sampled from the test set. The KL divergence is estimated using kernel density estimation over output dimensions for $q(f(\mathbf{X}_{\mathcal{E}};\boldsymbol{\theta}))$.

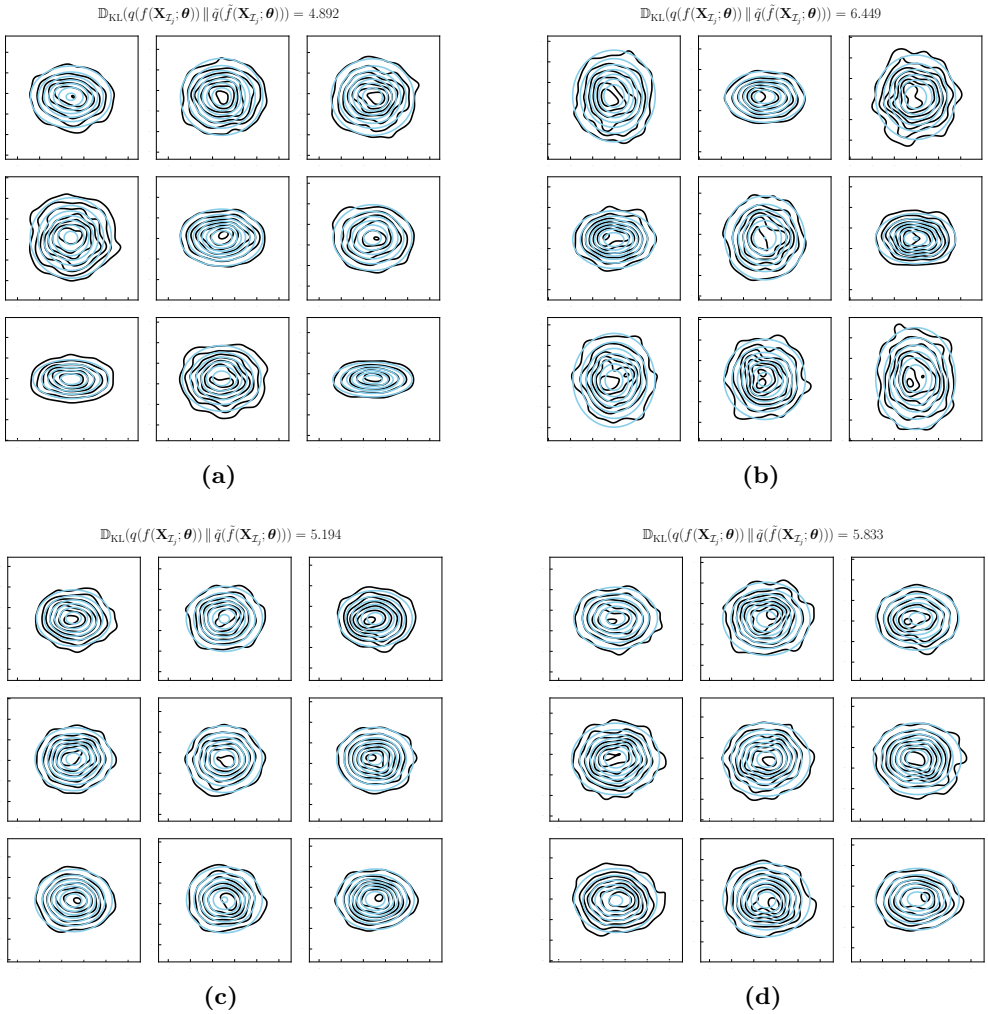
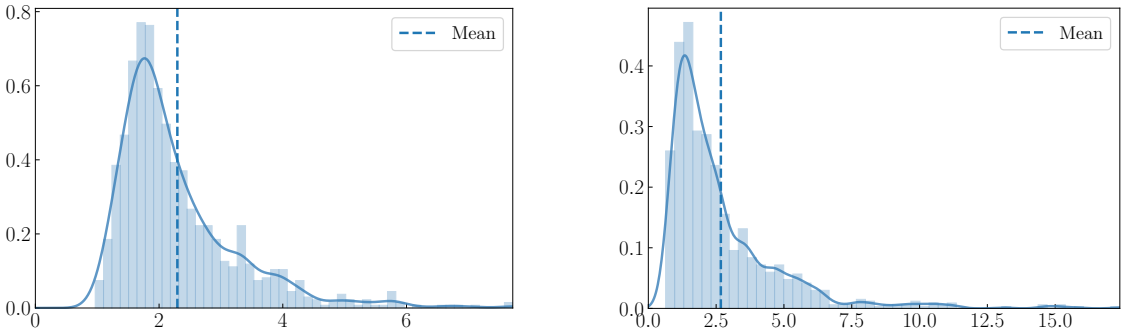


Figure 7: Distributions over functions under linearized and non-linearized mappings for a model trained on the FashionMNIST dataset. Each plot shows the covariance over stochastic functions (logits) between the first output dimension (corresponding to the first class) and all other output dimensions for a given input point sampled from the test set. The first output dimension is on the x-axis. The BNN's distribution over functions is shown in black, and the distribution over linearized functions is shown in light-blue. The title of each set of plots show the estimated KL divergence from the distribution over non-linearized functions to the distribution over linearized functions.

C.1.1 Validation of Linearization Assumption: CIFAR-10



(a) $\mathbb{D}_{\mathrm{KL}}(q(f(\mathbf{X}_{\mathcal{E}};\boldsymbol{\theta}))\,\|\,\tilde{q}(\tilde{f}(\mathbf{X}_{\mathcal{E}};\boldsymbol{\theta})))$ **(b)** $\mathbb{D}_{\mathrm{KL}}(\tilde{q}(\tilde{f}(\mathbf{X}_{\mathcal{E}};\boldsymbol{\theta}))\,\|\,q(f(\mathbf{X}_{\mathcal{E}};\boldsymbol{\theta})))$

Figure 8: KL divergences between distributions induced by $q_{\boldsymbol{\Theta}}$ under linearized and non-linearized mappings evaluated on 1,000 data points $\mathbf{X}_{\mathcal{E}}$ sampled from the test set. The KL divergence is estimated using kernel density estimation over output dimensions for $q(f(\mathbf{X}_{\mathcal{E}};\boldsymbol{\theta}))$.

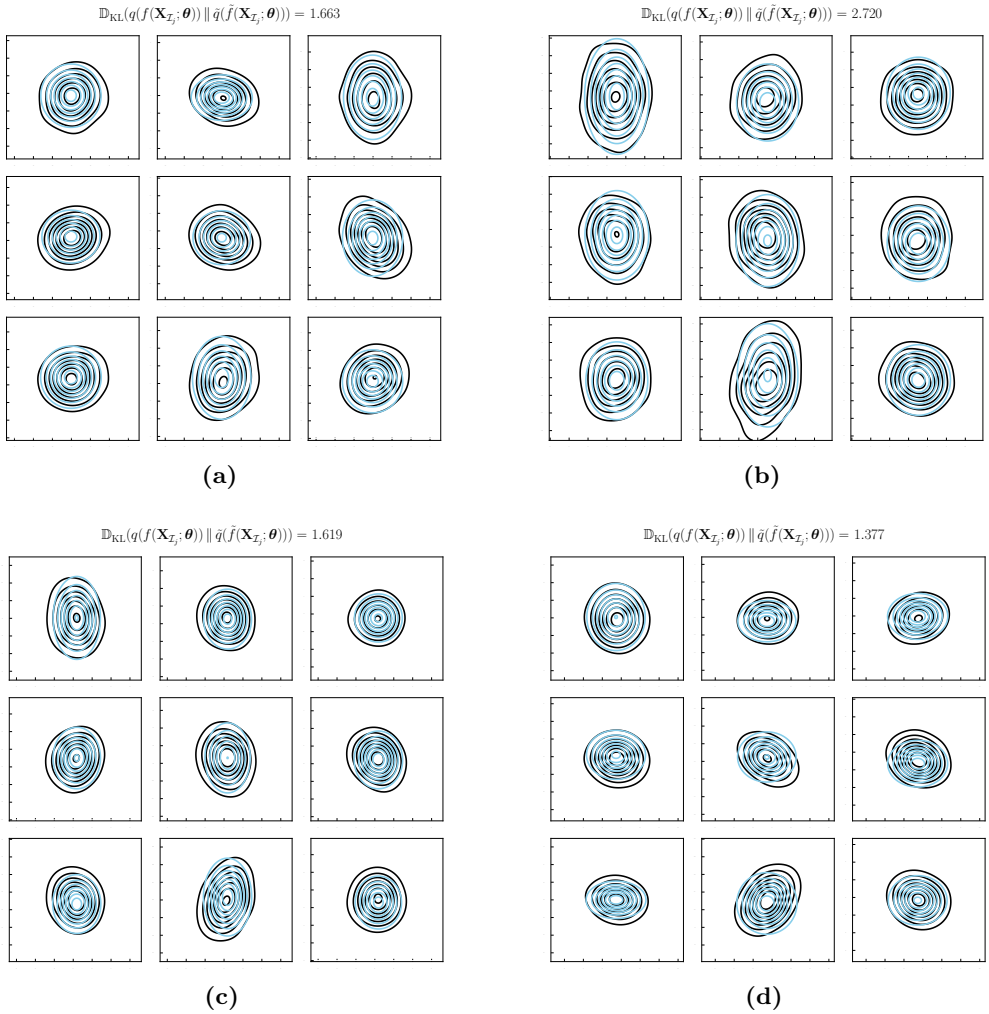


Figure 9: Distributions over functions under linearized and non-linearized mappings for a model trained on the CIFAR-10 dataset. Each plot shows the covariance over stochastic functions (logits) between the first output dimension (corresponding to the first class) and all other output dimensions for a given input point sampled from the test set. The first output dimension is on the x-axis. The BNN's distribution over functions is shown in black, and the distribution over linearized functions is shown in light-blue. The title of each set of plots show the estimated KL divergence from the distribution over non-linearized functions to the distribution over linearized functions.

Appendix D Further Empirical Results

D.1 FashionMNIST

Table 3: Comparison of in- and out-of-distribution performance metrics (mean ± standard error over ten random seeds). Best results are printed in boldface. For further details about model architectures and training, see Appendix B. AUROC for binary in- and out-of-distribution detection on MNIST/NotMNIST.

Dataset	Method	Accuracy ↑	NLL ↓	ECE ↓	OOD-AUROC (M/NM) ↑
FMNIST	MAP	91.73±0.08	0.288±0.003	0.037±0.001	87.00±0.30 / 74.85±1.31
	MFVI	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)	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
	DUQ (van Amersfoort et al., 2020)	92.40±0.20	–	–	95.50±0.70 / 94.60±1.80
	BNN-GLM (Immer et al., 2020)	92.25±0.10	0.244±0.003	0.012±0.003	95.55±0.60 / –
	FSVI	**93.15**±0.13	**0.203**±0.004	0.014±0.002	**96.55**±0.41 / **95.27**±0.63
	FSVI (MAP init)	90.46±0.06	0.299±0.003	**0.009**±0.001	93.40±0.42 / 92.19±0.39
	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
	Deep Ensemble	92.49±0.01	0.242±0.001	**0.019**±0.000	89.22±0.09 / 83.17±0.91
	MC DROPOUT Ensemble	92.30±0.03	0.221±0.000	0.019±0.001	90.17±0.29 / 79.70±0.76
	MFVI Ensemble	92.46±0.04	0.294±0.001	0.026±0.000	94.29±0.21 / 90.31±0.37
	MFVI (tempered) Ensemble	92.21±0.03	0.398±0.002	0.040±0.001	89.46±0.26 / 82.19±0.29
	FSVI Ensemble	**94.44**±0.07	**0.181**±0.001	0.020±0.001	**97.85**±0.15 / **96.95**±0.20
	FSVI (MAP init) Ensemble	92.45±0.05	0.242±0.001	0.019±0.001	96.06±0.23 / 94.89±0.20

D.2 MNIST

Table 4: Comparison of in- and out-of-distribution performance metrics (mean ± standard error over ten random seeds). Best results are printed in boldface. For further details about model architectures and training, see Appendix B. AUROC for binary in- and out-of-distribution detection on NotMNIST/FashionMNIST.

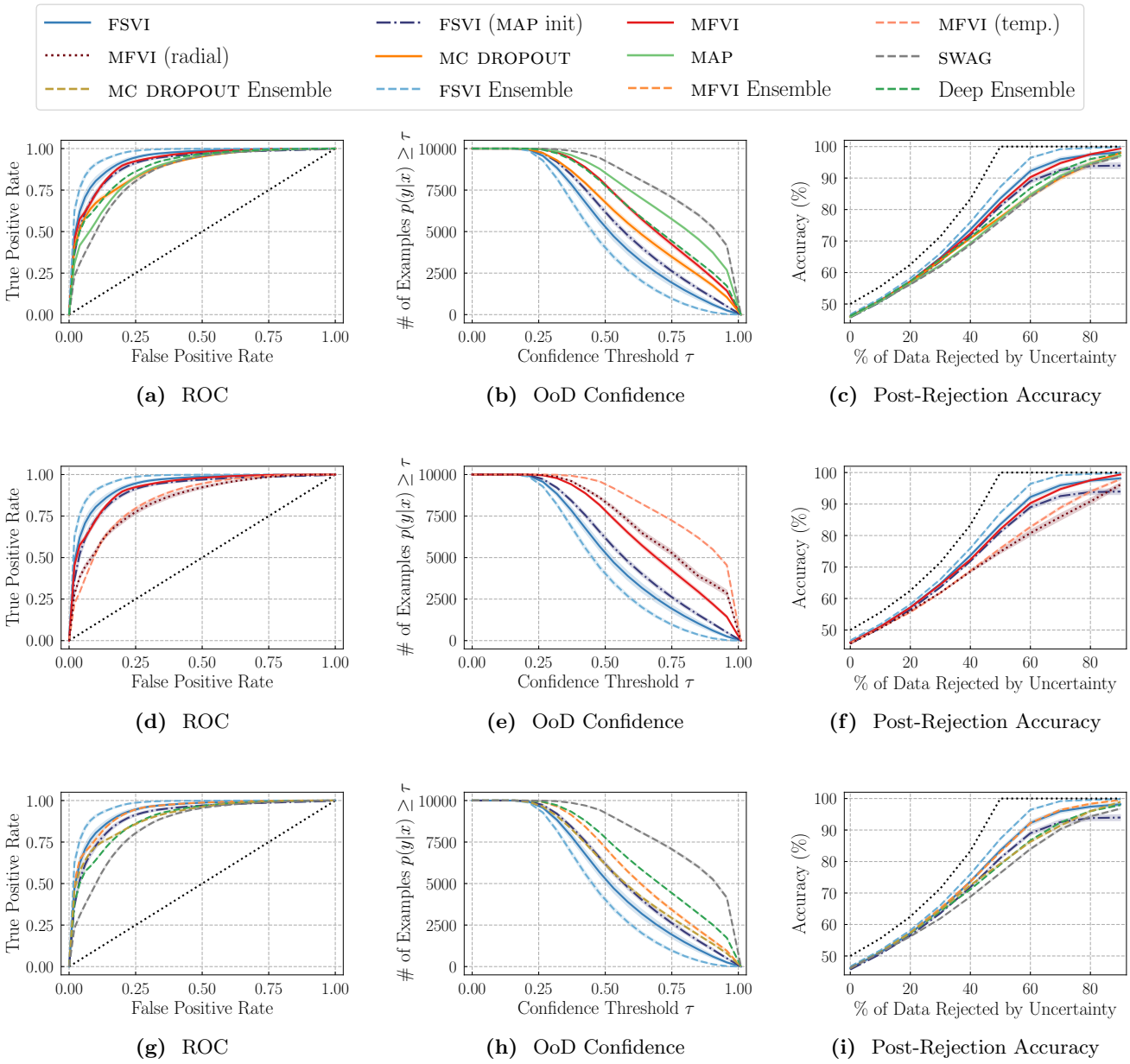
Dataset	Method	Accuracy ↑	NLL ↓	ECE ↓	OOD-AUROC (NM/FM)† ↑
MNIST	MAP	97.84±0.04	0.069±0.001	0.003±0.000	88.13±1.02 / 94.90±0.57
	MFVI	97.28±0.03	0.060±0.001	0.006±0.000	93.74±0.91 / **96.73**±0.24
	MFVI (tempered)	97.74±0.03	0.088±0.002	0.011±0.000	91.00±1.08 / 93.67±0.45
	MC DROPOUT	97.48±0.04	0.068±0.001	0.010±0.001	88.63±1.25 / 96.11±0.17
	FSVI	97.47±0.11	0.087±0.003	0.009±0.000	**98.99**±0.33 / 96.67±0.33
	Deep Ensemble	98.41±0.01	0.054±0.000	0.011±0.000	95.02±0.20 / 97.00±0.13
	MC DROPOUT Ensemble	98.32±0.03	0.061±0.001	0.016±0.000	93.34±0.37 / 96.88±0.07
	MFVI Ensemble	98.46±0.02	0.052±0.000	0.007±0.000	97.27±0.23 / 96.50±0.17
	MFVI (tempered) Ensemble	98.32±0.02	0.056±0.001	0.003±0.000	96.39±0.29 / 95.82±0.12
	FSVI Ensemble	98.16±0.01	0.060±0.001	0.004±0.000	**99.71**±0.04 / **97.91**±0.10

D.3 CIFAR-10

Table 5: Comparison of in- and out-of-distribution performance metrics (mean ± standard error over ten random seeds). Best results are printed in boldface. For further details about model architectures and training, see Appendix B. AUROC for binary in- and out-of-distribution detection on SVHN.

Dataset	Method	Accuracy ↑	NLL ↓	ECE ↓	OOD-AUROC/C-CIFAR Acc. ↑
			CNN		
CIFAR-10	MAP	87.35±0.09	0.491±0.005	0.070±0.001	90.86±0.43 / 74.20±0.60
	MFVI	84.04±0.07	0.372±0.002	**0.016**±0.001	92.62±0.31 / 71.48±0.74
	MFVI (tempered)	86.29±0.08	0.457±0.003	0.049±0.001	91.54±0.57 / 72.02±0.58
	MFVI (radial)	83.99±0.18	0.510±0.001	0.048±0.002	86.04±0.18 / 73.54±0.53
	MC DROPOUT	83.89±0.18	0.412±0.005	0.018±0.002	92.69±0.49 / 69.75±0.82
	BNN-GLM (Immer et al., 2020)	81.37±0.15	0.601±0.008	0.084±0.010	84.30±0.02 / –
	FSVI	86.34±0.11	0.499±0.005	0.061±0.001	94.00±0.39 / 73.59±0.66
	FSVI (MAP init)	**88.10**±0.08	**0.330**±0.003	0.021±0.001	**97.71**±0.11 / **79.64**±0.36
	SWAG (Maddox et al., 2019)	89.73±0.14	0.480±0.001	0.067±0.002	89.79±0.50 / 76.12±0.51
	Deep Ensemble	89.28±0.04	0.339±0.003	0.020±0.001	92.00±0.16 / 76.65±0.21
	MC DROPOUT Ensemble	88.02±0.09	0.371±0.002	0.056±0.001	91.92±0.14 / 72.89±0.57
	MFVI Ensemble	89.49±0.07	0.330±0.001	0.049±0.001	94.06±0.20 / 73.67±0.38
	MFVI (tempered) Ensemble	89.78±0.04	0.321±0.001	0.014±0.001	92.07±0.40 / 75.07±0.26
	FSVI Ensemble	**90.17**±0.03	**0.314**±0.001	**0.018**±0.001	**96.17**±0.10 / **78.63**±0.40
			ResNet-18		
	MAP	92.19±0.15	0.307±0.006	0.046±0.001	95.17±0.40 / 78.55±1.01
	MFVI	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	0.31±0.004	0.041±0.001	90.32±0.57 / 80.19±1.44
	DUQ (van Amersfoort et al., 2020)	94.10±0.2	–	–	92.70±1.30 / –
	VOGN (Osawa et al., 2019)	84.27±0.20	0.477±0.006	0.040±0.002	87.60±0.20 / –
	FSVI	93.30±0.04	0.295±0.003	0.034±0.001	94.89±0.19 / 80.88±0.47
	FSVI (MAP init)	**94.10**±0.04	**0.175**±0.002	**0.014**±0.001	**98.89**±0.23 / **81.38**±0.41
	Deep Ensemble	95.13±0.06	0.158±0.001	0.019±0.001	98.04±0.07 / 81.22±0.37
	FSVI Ensemble	**95.29**±0.03	**0.150**±0.003	**0.011**±0.001	**99.12**±0.36 / **81.44**±0.43

D.4 Out-of-Distribution Performance FashionMNIST/MNIST



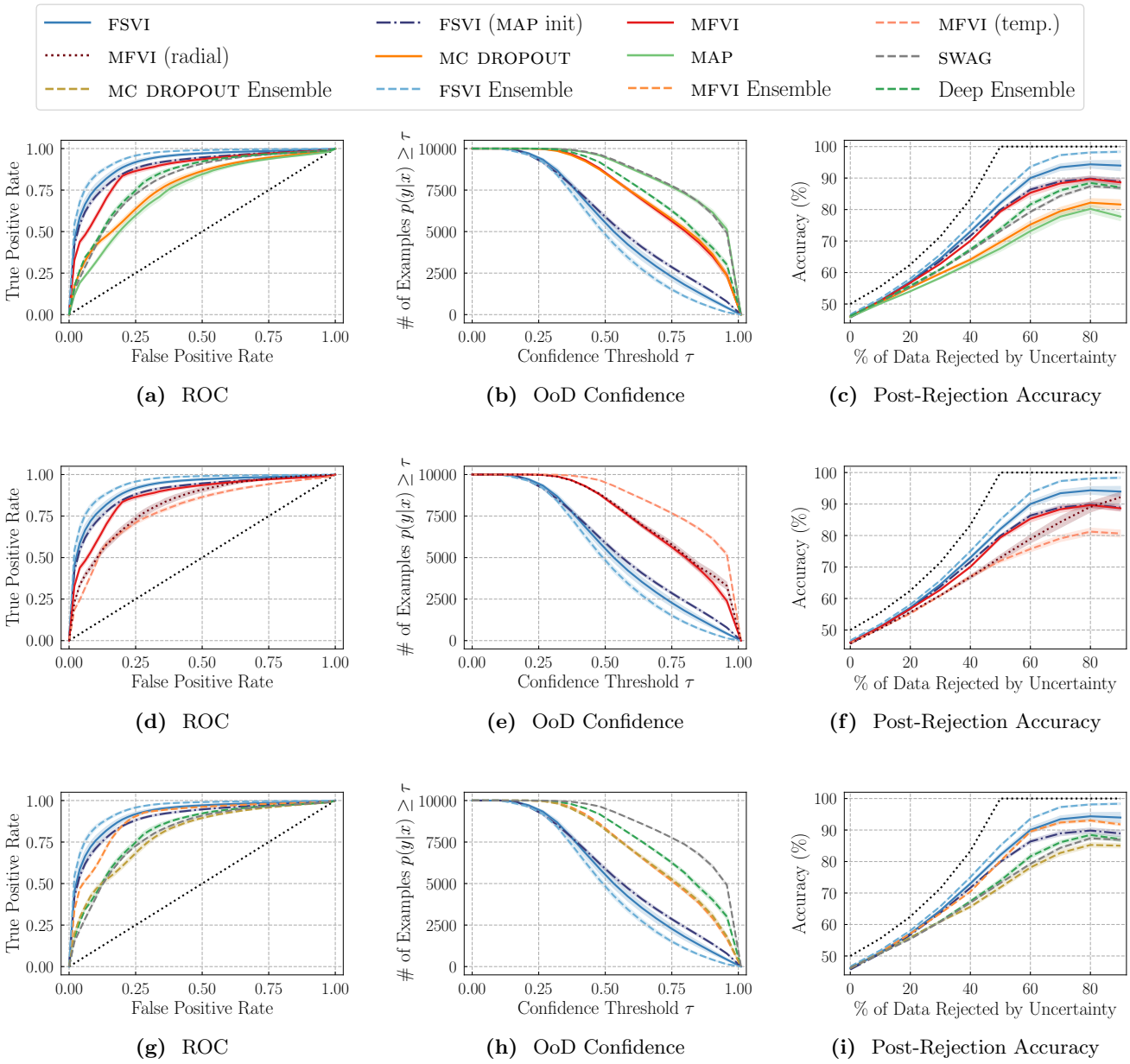
(a) ROC **(b)** OoD Confidence **(c)** Post-Rejection Accuracy

(d) ROC **(e)** OoD Confidence **(f)** Post-Rejection Accuracy

(g) ROC **(h)** OoD Confidence **(i)** Post-Rejection Accuracy

Figure 10: Uncertainty evaluation metrics for out-of-distribution prediction. Models were trained on FashionMNIST and MNIST is used as out-of-distribution data. Shading denotes the standard error, computed over ten random seeds. **Left**: Receiver operating characteristic for out-of-distribution detection. Curves closer to the top left are better. **Center**: Model confidence on out-of-distribution inputs. Curves closer to the bottom left are better. **Right**: Accuracy after rejecting X% of evaluation samples with the highest predictive uncertainty. Curves closer to the theoretical maximum (denoted by the dotted line) are better. The figures show that FSVI consistently outperforms related methods in terms of classifying in- and out-of-distribution datapoints (left), generating low-confidence predictions on out-of-distribution data (center), and using predictive uncertainty to identify points where the model's predictions would be incorrect (right).

D.5 Out-of-Distribution Performance FashionMNIST/NotMNIST



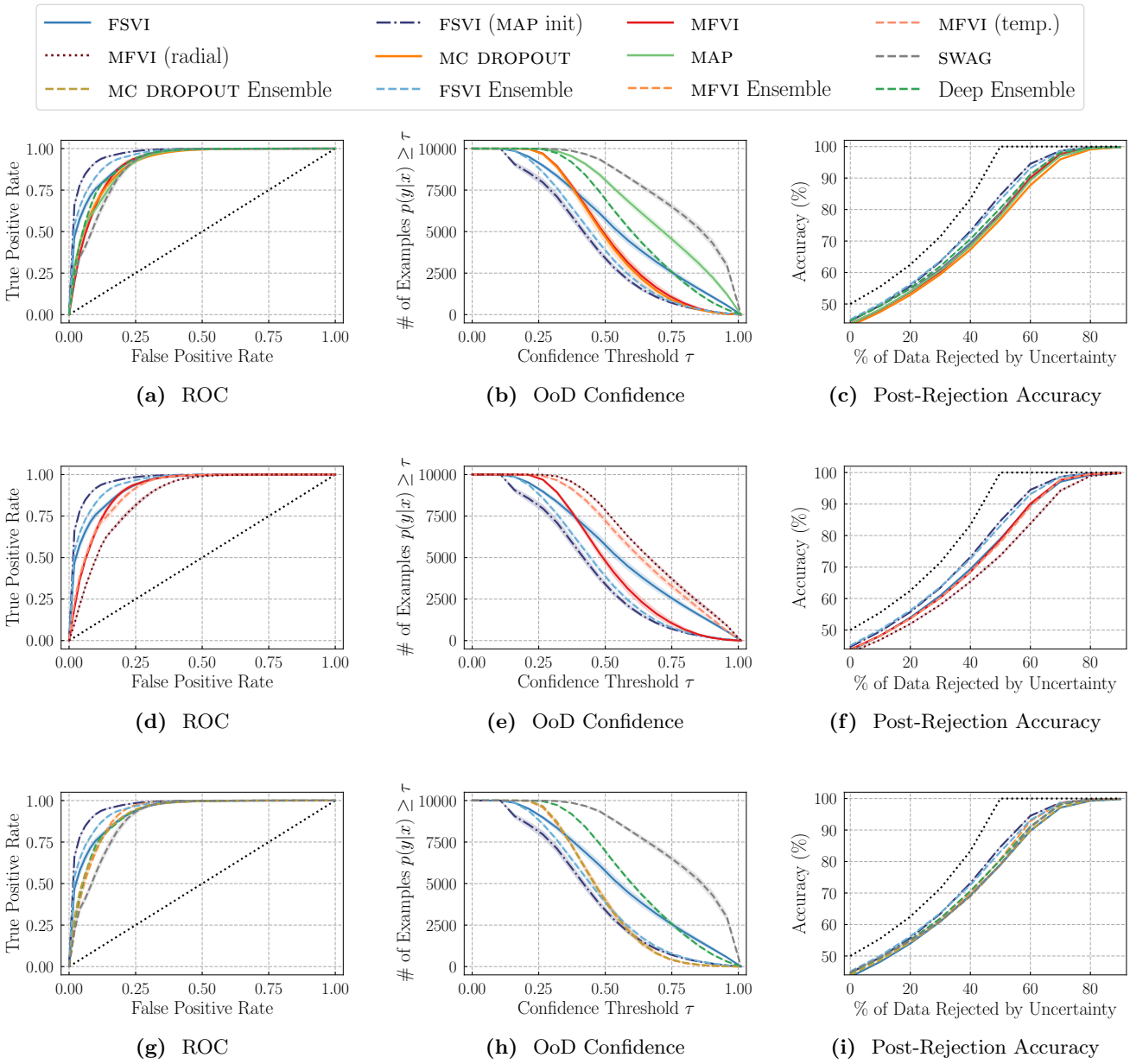
(a) ROC **(b)** OoD Confidence **(c)** Post-Rejection Accuracy

(d) ROC **(e)** OoD Confidence **(f)** Post-Rejection Accuracy

(g) ROC **(h)** OoD Confidence **(i)** Post-Rejection Accuracy

Figure 11: Uncertainty evaluation metrics for out-of-distribution prediction. Models were trained on FashionMNIST and NotMNIST is used as out-of-distribution data. Shading denotes the standard error, computed over ten random seeds. **Left**: Receiver operating characteristic for out-of-distribution detection. Curves closer to the top left are better. **Center**: Model confidence on out-of-distribution inputs. Curves closer to the bottom left are better. **Right**: Accuracy after rejecting X% of evaluation samples with the highest predictive uncertainty. Curves closer to the theoretical maximum (denoted by the dotted line) are better. The figures show that FSVI consistently outperforms related methods in terms of classifying in- and out-of-distribution datapoints (left), generating low-confidence predictions on out-of-distribution data (center), and using predictive uncertainty to identify points where the model's predictions would be incorrect (right).

D.6 Out-of-Distribution Performance CIFAR-10/SVHN



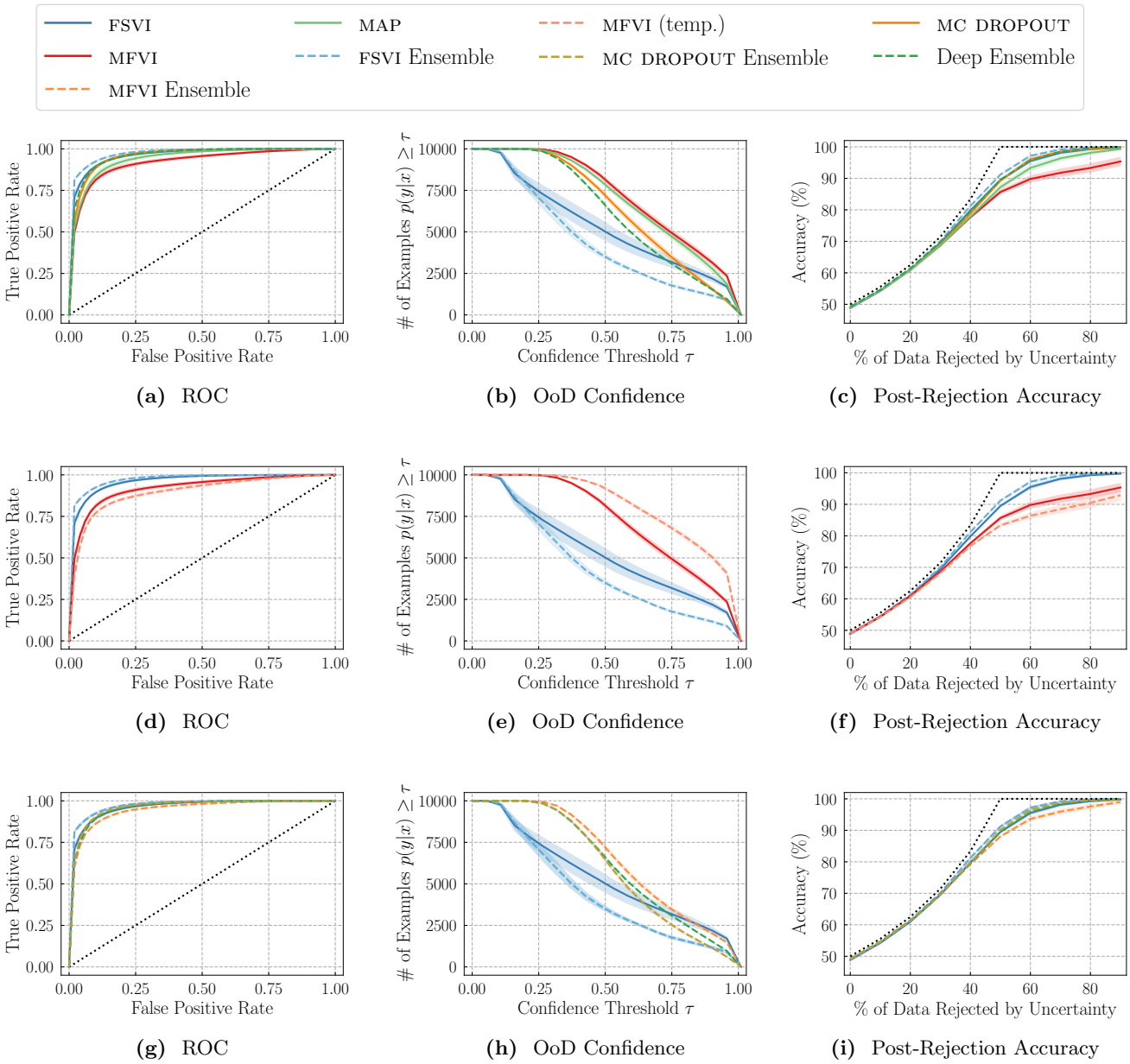
(a) ROC **(b)** OoD Confidence **(c)** Post-Rejection Accuracy

(d) ROC **(e)** OoD Confidence **(f)** Post-Rejection Accuracy

(g) ROC **(h)** OoD Confidence **(i)** Post-Rejection Accuracy

Figure 12: Uncertainty evaluation metrics for out-of-distribution prediction. Models were trained on CIFAR-10 and SVHN is used as out-of-distribution data. Shading denotes the standard error, computed over ten random seeds. **Left**: Receiver operating characteristic for out-of-distribution detection. Curves closer to the top left are better. **Center**: Model confidence on out-of-distribution inputs. Curves closer to the bottom left are better. **Right**: Accuracy after rejecting X% of evaluation samples with the highest predictive uncertainty. Curves closer to the theoretical maximum (denoted by the dotted line) are better. The figures show that FSVI consistently outperforms related methods in terms of classifying in- and out-of-distribution datapoints (left), generating low-confidence predictions on out-of-distribution data (center), and using predictive uncertainty to identify points where the model's predictions would be incorrect (right).

D.7 Out-of-Distribution Performance MNIST/FashionMNIST



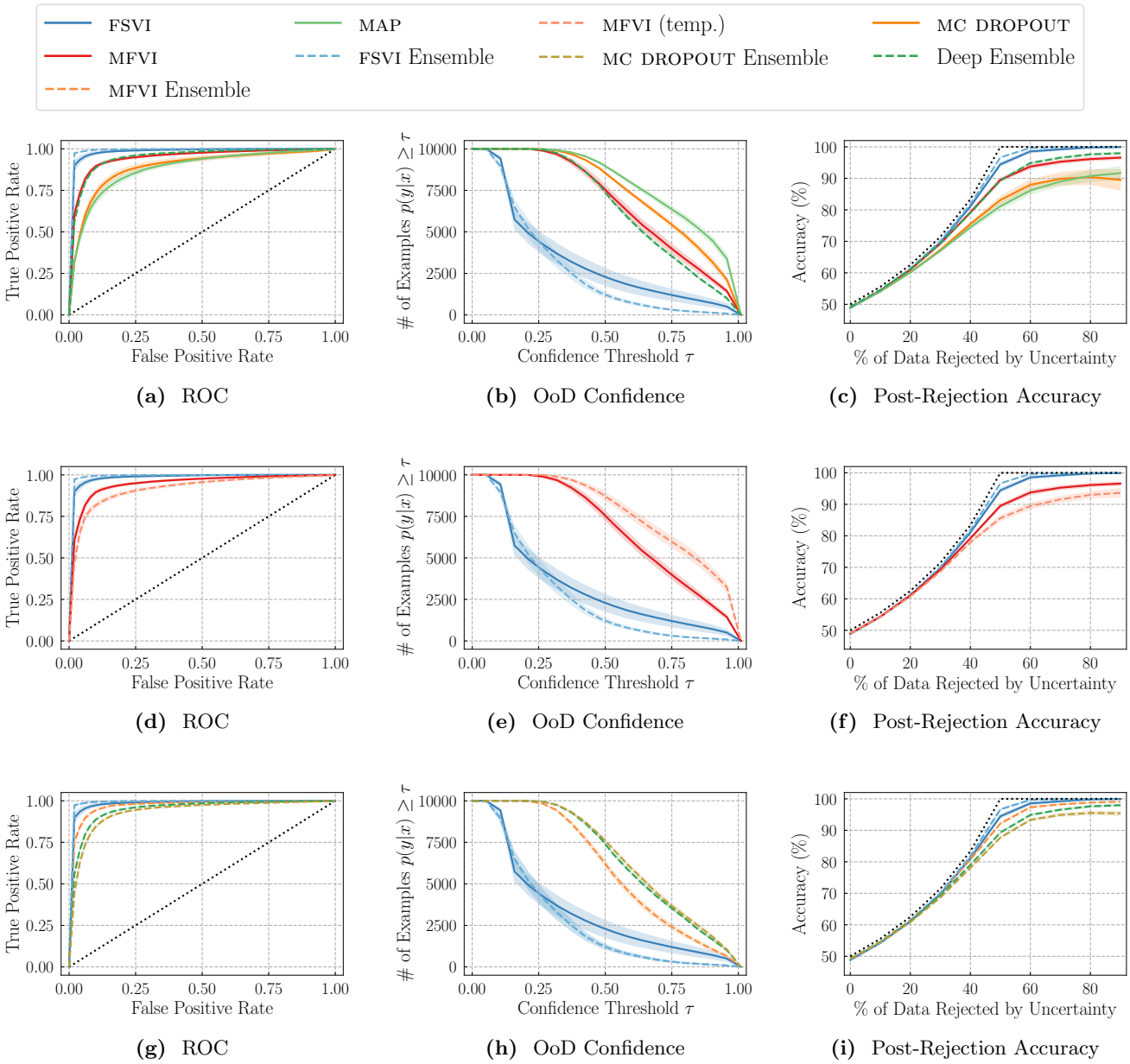
(a) ROC **(b)** OoD Confidence **(c)** Post-Rejection Accuracy

(d) ROC **(e)** OoD Confidence **(f)** Post-Rejection Accuracy

(g) ROC **(h)** OoD Confidence **(i)** Post-Rejection Accuracy

Figure 13: Uncertainty evaluation metrics for out-of-distribution prediction. Models were trained on MNIST and FashionMNIST is used as out-of-distribution data. Shading denotes the standard error, computed over ten random seeds. **Left**: Receiver operating characteristic for out-of-distribution detection. Curves closer to the top left are better. **Center**: Model confidence on out-of-distribution inputs. Curves closer to the bottom left are better. **Right**: Accuracy after rejecting X% of evaluation samples with the highest predictive uncertainty. Curves closer to the theoretical maximum (denoted by the dotted line) are better. The figures show that FSVI consistently outperforms related methods in terms of classifying in- and out-of-distribution datapoints (left), generating low-confidence predictions on out-of-distribution data (center), and using predictive uncertainty to identify points where the model's predictions would be incorrect (right).

D.8 Out-of-Distribution Performance MNIST/NotMNIST



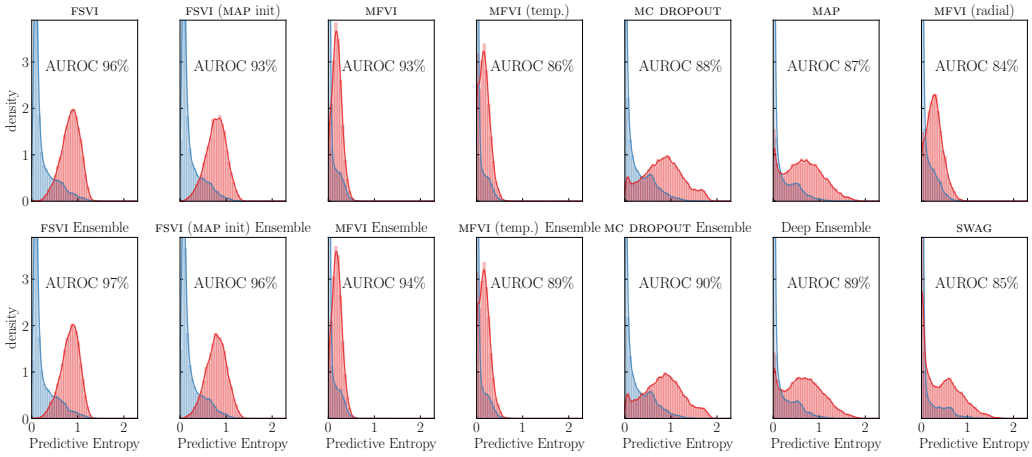
(a) ROC **(b)** OoD Confidence **(c)** Post-Rejection Accuracy

(d) ROC **(e)** OoD Confidence **(f)** Post-Rejection Accuracy

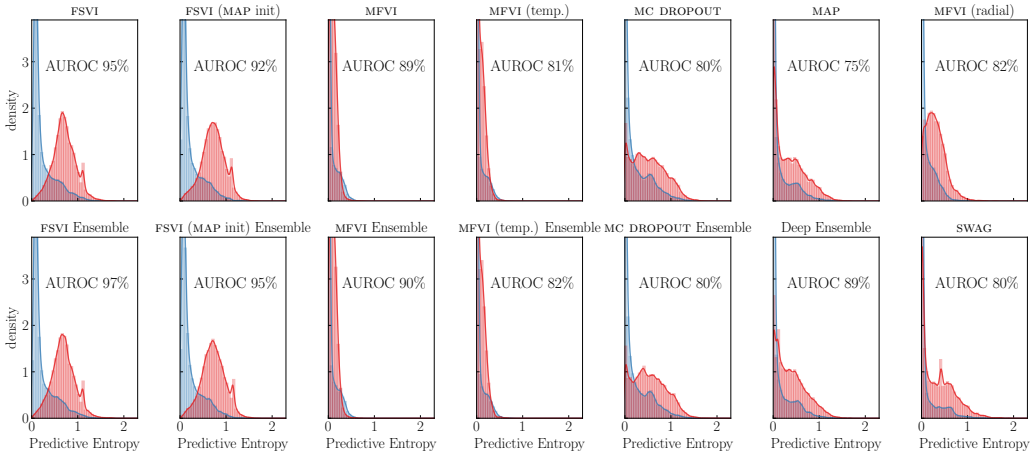
(g) ROC **(h)** OoD Confidence **(i)** Post-Rejection Accuracy

Figure 14: Uncertainty evaluation metrics for out-of-distribution prediction. Models were trained on MNIST and NotMNIST is used as out-of-distribution data. Shading denotes the standard error, computed over ten random seeds. **Left**: Receiver operating characteristic for out-of-distribution detection. Curves closer to the top left are better. **Center**: Model confidence on out-of-distribution inputs. Curves closer to the bottom left are better. **Right**: Accuracy after rejecting X% of evaluation samples with the highest predictive uncertainty. Curves closer to the theoretical maximum (denoted by the dotted line) are better. The figures show that FSVI consistently outperforms related methods in terms of classifying in- and out-of-distribution datapoints (left), generating low-confidence predictions on out-of-distribution data (center), and using predictive uncertainty to identify points where the model's predictions would be incorrect (right).

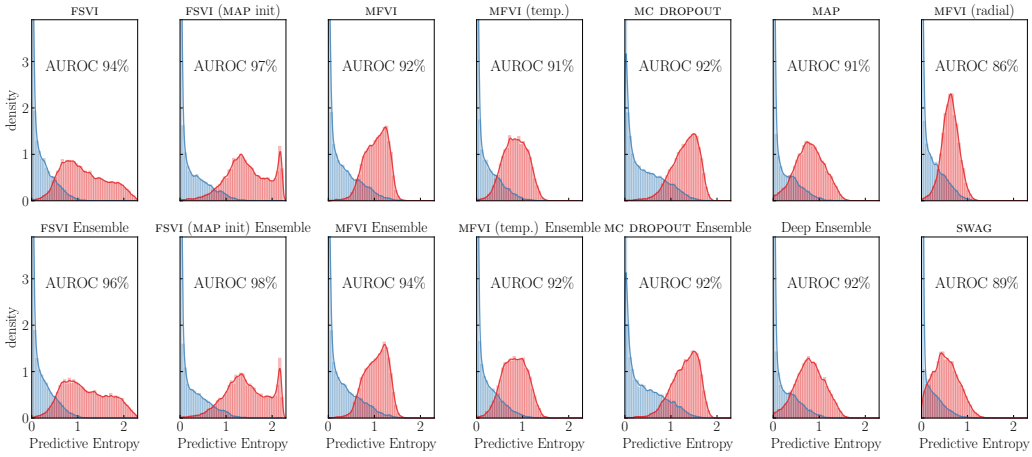
D.9 Predictive Entropy on In- and Out-of-Distribution Inputs



(a) In-Distribution: FashionMNIST, **Out-of-Distribution**: MNIST

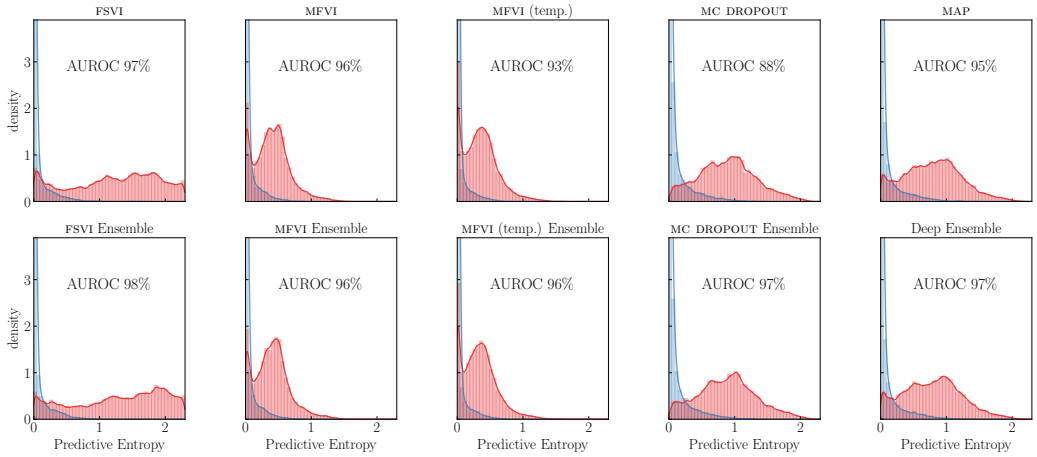


(b) In-Distribution: FashionMNIST, **Out-of-Distribution**: NotMNIST

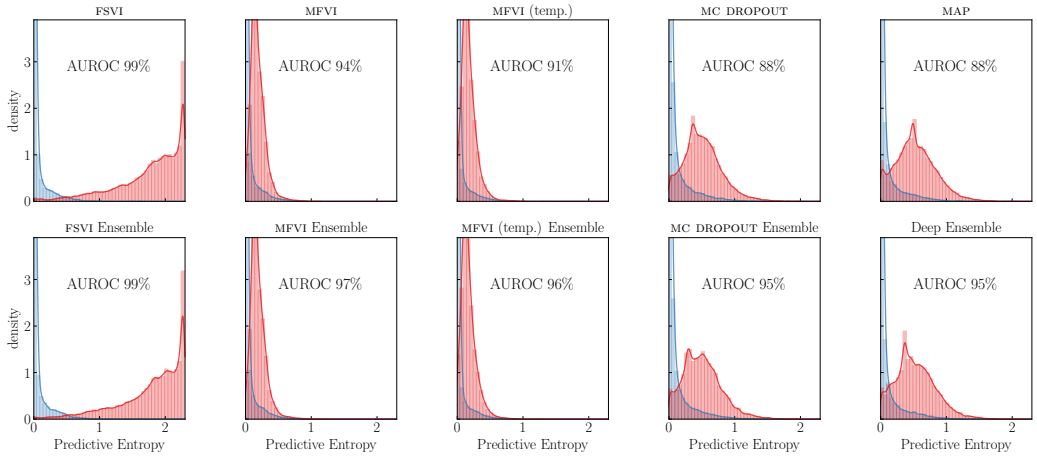


(c) In-Distribution:CIFAR-10, **Out-of-Distribution**: SVHN

Figure 15: Histograms of predictive entropy estimates. Low predictive entropy corresponds to low predictive uncertainty and high predictive entropy corresponds to high predictive uncertainty. Models with reliable predictive uncertainty estimates would exhibit low predictive entropy on in-distribution inputs (i.e., a predictive entropy distribution concentrated near zero) and high predictive entropy on out-of-distribution inputs (i.e., a predictive entropy distribution concentrated away from zero) with little overlap between the distributions.



(a) In-Distribution: MNIST, **Out-of-Distribution**: FashionMNIST

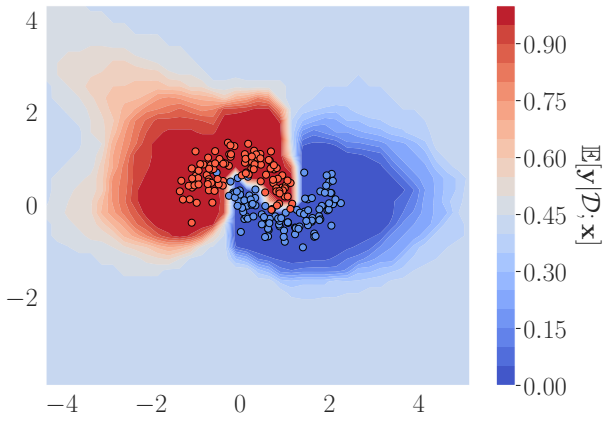


(b) In-Distribution: MNIST, **Out-of-Distribution**: NotMNIST

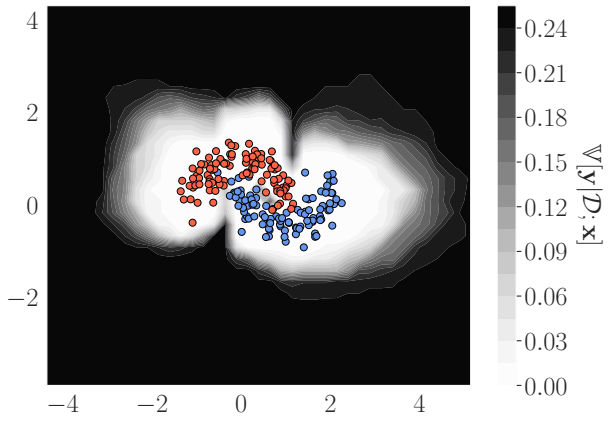
Figure 16: Histograms of predictive entropy estimates. Low predictive entropy corresponds to low predictive uncertainty and high predictive entropy corresponds to high predictive uncertainty. Models with reliable predictive uncertainty estimates would exhibit low predictive entropy on in-distribution inputs (i.e., a predictive entropy distribution concentrated near zero) and high predictive entropy on out-of-distribution inputs (i.e., a predictive entropy distribution concentrated away from zero) with little overlap between the distributions.

Appendix E Illustrative Examples

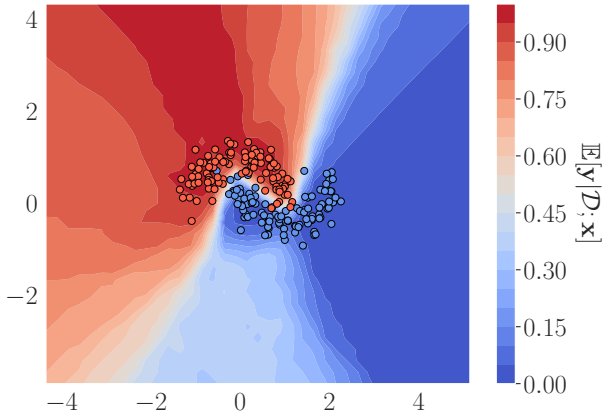
E.1 *Two Moons* Classification Dataset



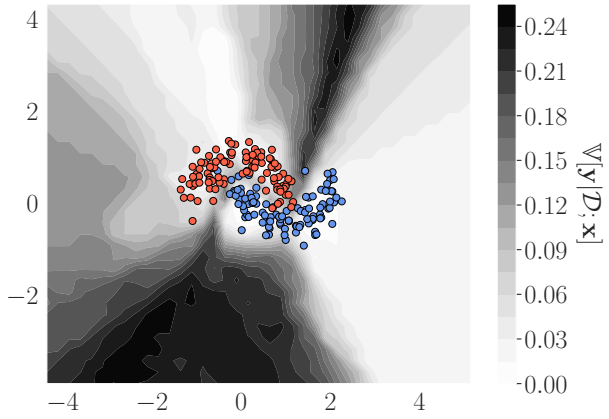
(a) FSVI: Posterior Predictive Mean



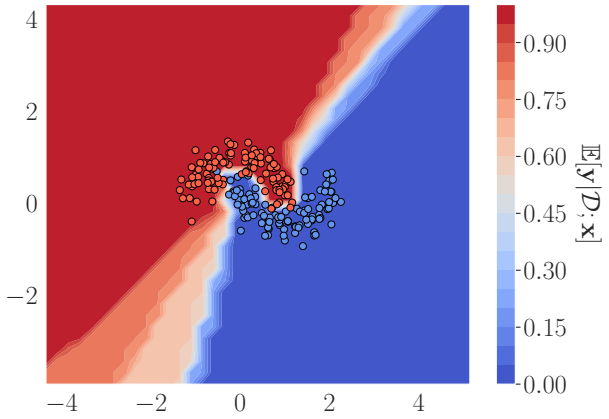
(b) FSVI: Posterior Predictive Variance



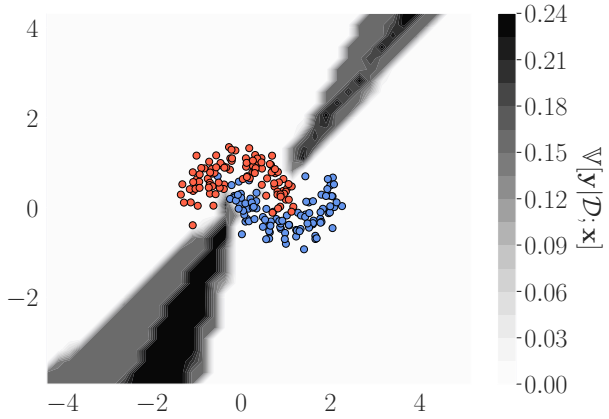
(c) MFVI: Posterior Predictive Mean



(d) MFVI: Posterior Predictive Variance



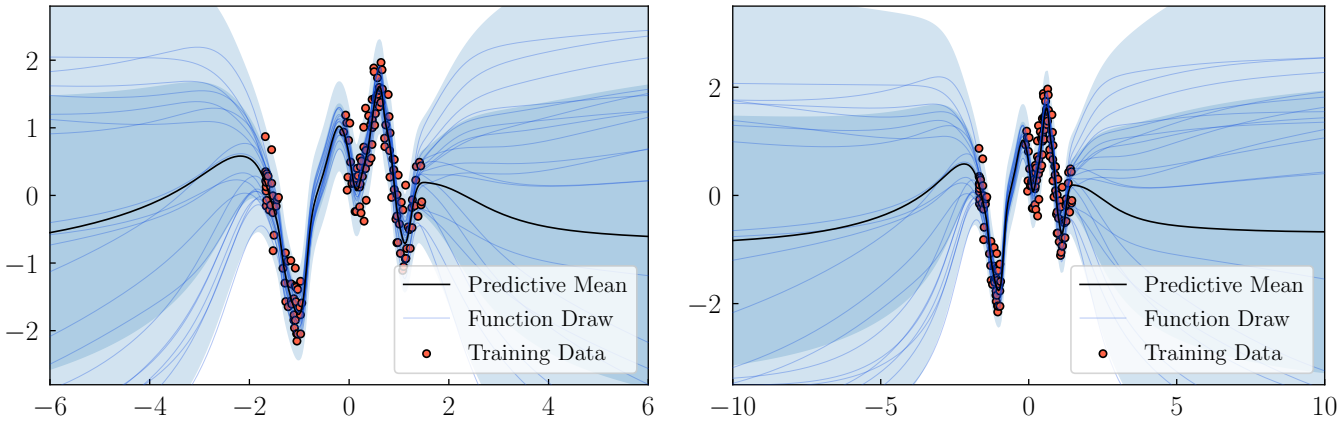
(e) MAP Ensemble: Predictive Mean



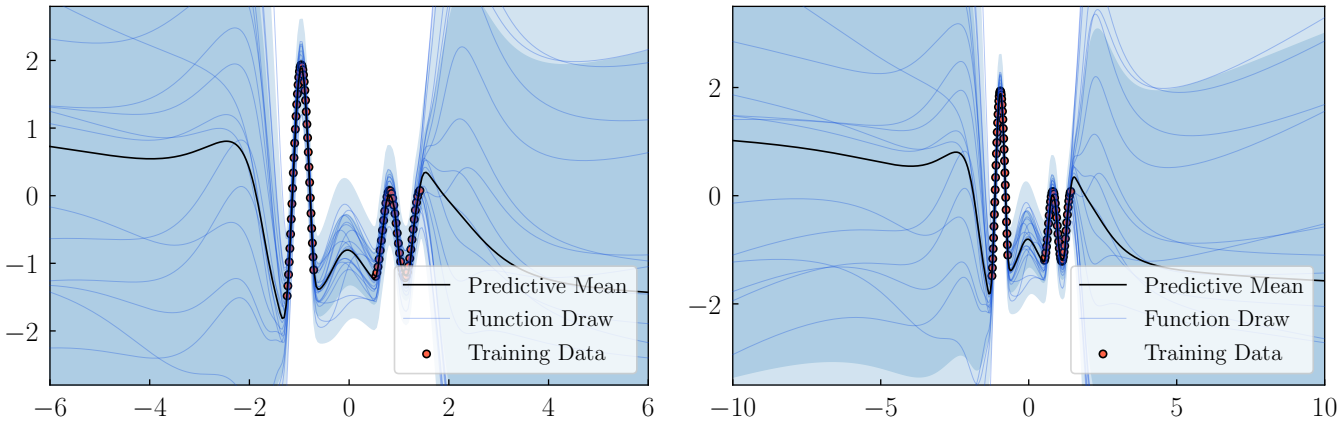
(f) MAP Ensemble: Predictive Variance

Figure 17: Binary classification on the *Two Moons* dataset. The plots show the posterior predictive mean and variance of a BNN trained via FSVI (Figure 17a and Figure 17b), of a BNN trained via MFVI (Figure 17c and Figure 17d), and an ensemble of MAP models (Figure 17e and Figure 17f). 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.

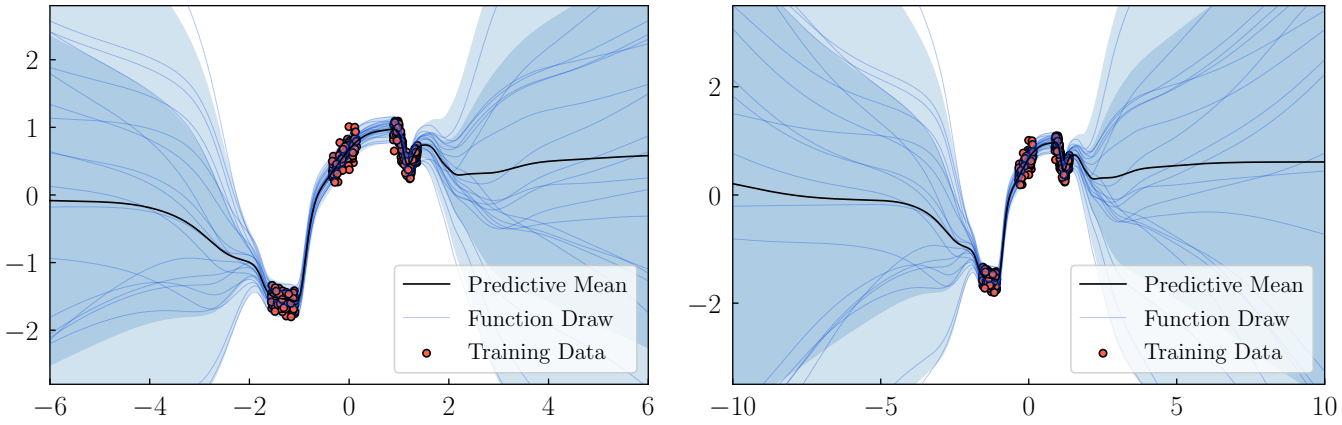
E.2 Illustrative Examples: 1D Regression



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



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



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

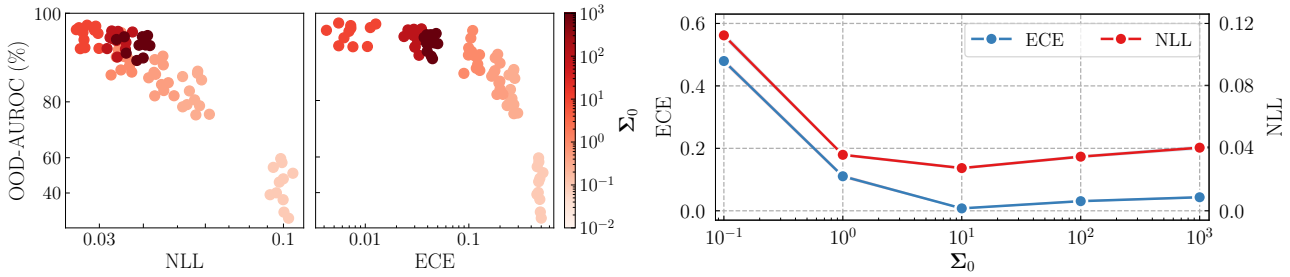
Figure 18: 1D Regression with FSVI on a selection of datasets used to demonstrate desirable predictive uncertainty estimates in prior works. The left column is zoomed in.

Appendix F Ablation Studies

We performed a comprehensive set of ablation studies to understand 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 inducing samples used per gradient steps, the inducing input selection method, 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.

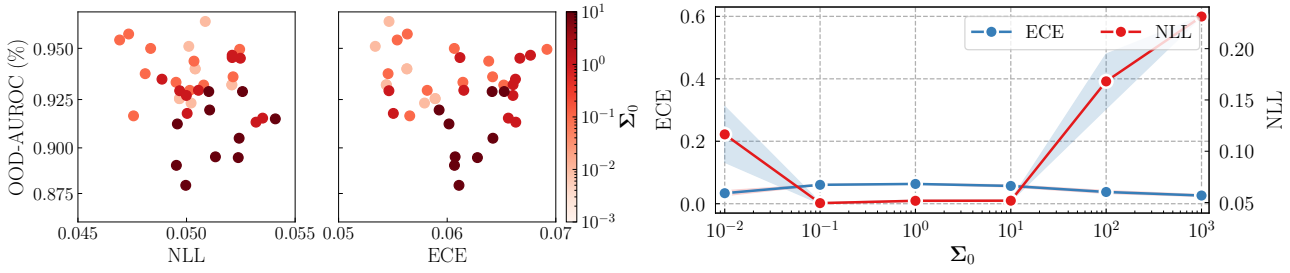
F.1 Ablation Study on the Effect of Different Function-Space Priors

To better understand FSVI, we investigate what hyperparameter choice leads to good predictive performance and reliable out-of-distribution predictive uncertainty. Figure 19 and Figure 20 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 19b, test ECE is lowest for a prior variance of $\mathbf{\Sigma}_0 = 10$, while Figure 19a 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 FSVI in Tables 3, 4, and 5. For further ablations and details on hyperparameter selection, see Appendix D.



(a) Effect of Prior Variance $\mathbf{\Sigma}_0$ **(b)** ECE and NLL vs. $\mathbf{\Sigma}_0$

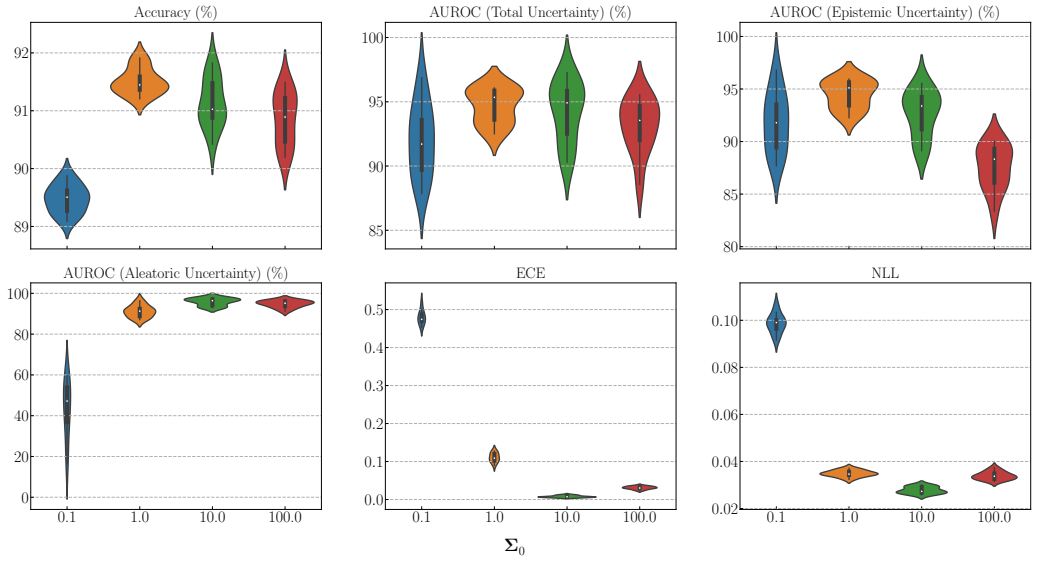
Figure 19: For BNNs trained via FSVI on FashionMNIST, the figures show the relationship between OOD-AUROC (on SVHN), the negative log-likelihood (NLL), expected calibration error (ECE), and the prior variance. OOD-AUROC is very negatively correlated with both NLL and ECE. All metrics are optimal at a prior covariance of $\mathbf{\Sigma}_0 = 10$. For further results, see Appendix D.



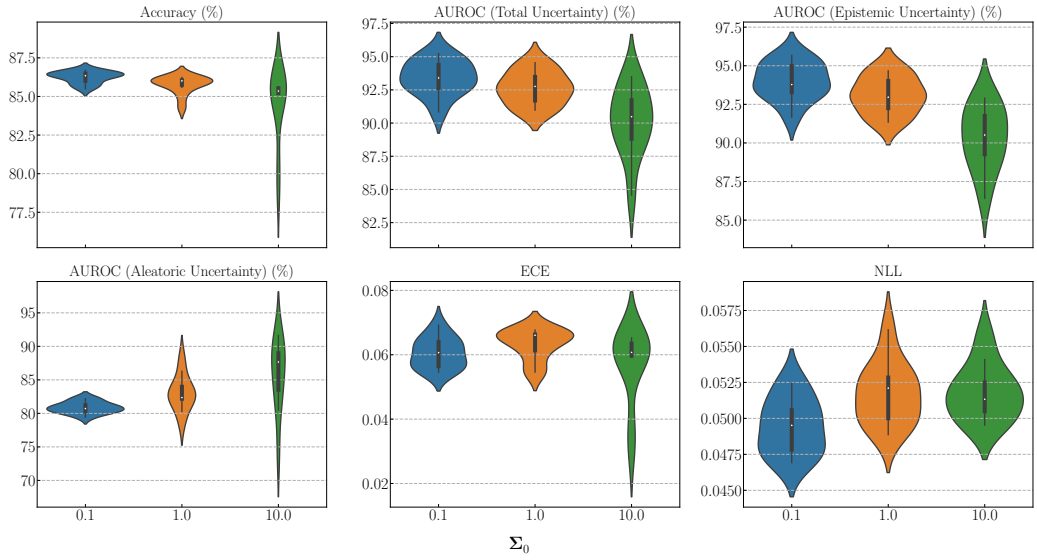
(a) Effect of Prior Variance $\mathbf{\Sigma}_0$ **(b)** ECE and NLL vs. $\mathbf{\Sigma}_0$

Figure 20: 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 very negatively correlated with both NLL and ECE. All metrics are optimal at a prior covariance of $\mathbf{\Sigma}_0 = 10$. For further results, see Appendix D.

F.2 Prior Variance



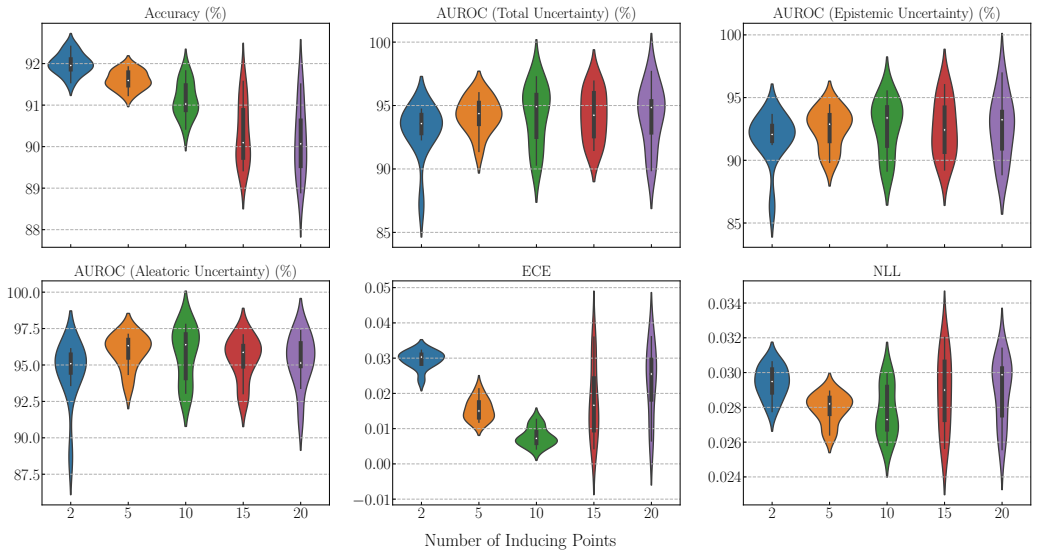
(a) FashionMNIST



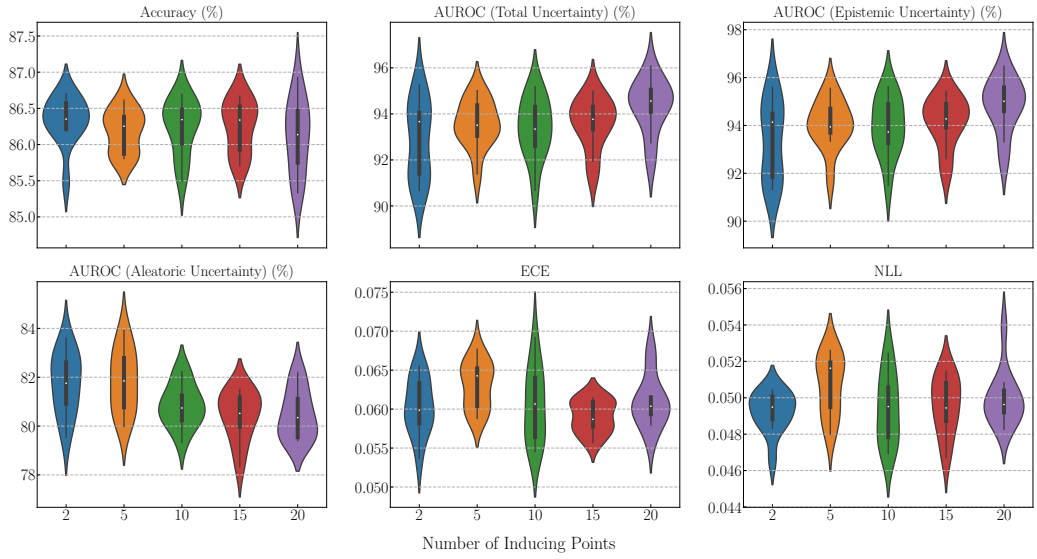
(b) CIFAR-10

Figure 21: Effect of prior variance over parameters $\mathbf{\Sigma}_0$ on in- and out-of-distribution evaluation metrics. For all experiments, 10 inducing inputs were sampled for each gradient step. For FashionMNIST, a fraction of 50% of inducing inputs were sampled from the training set, while for CIFAR-10 no inducing inputs were sampled from the training set. 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.

F.3 Number of Inducing Inputs



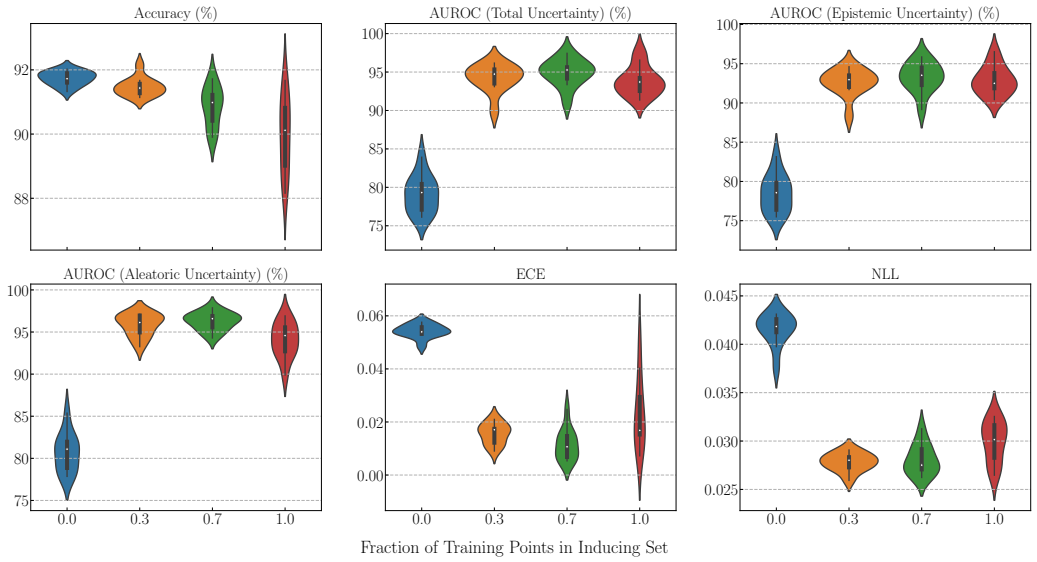
(a) FashionMNIST



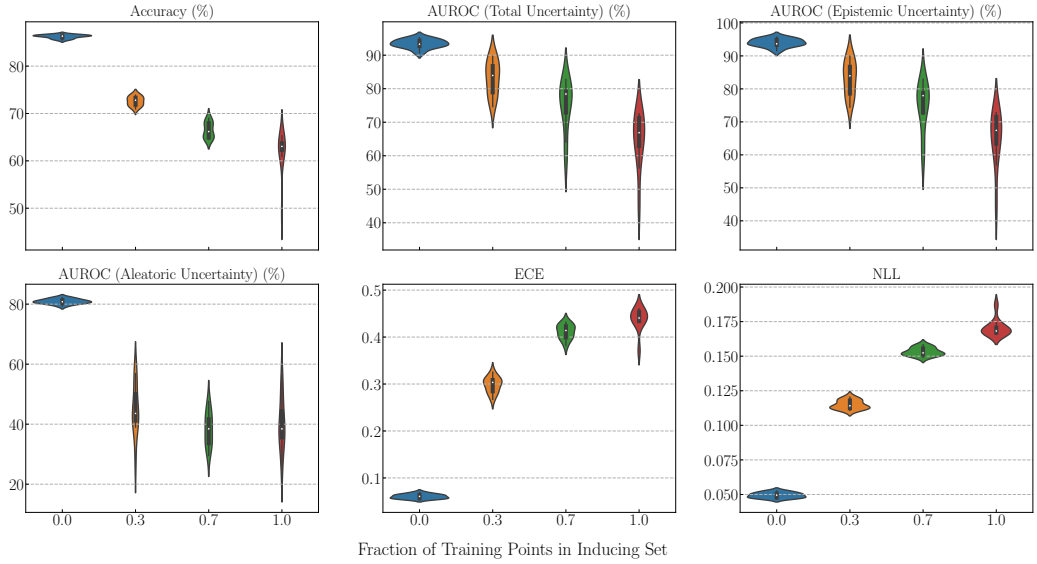
(b) CIFAR-10

Figure 22: Effect of the number of inducing points used to evaluate the KL divergence at each gradient step on in- and out-of-distribution evaluation metrics. For FashionMNIST, a fraction of 50% of inducing inputs were sampled from the training set, while for CIFAR-10 no inducing inputs were sampled from the training set. A prior variance of 10 was used for FashionMNIST and a prior variance of 0.1 was used for CIFAR-10. 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.

F.4 Inducing Input Sampling Method



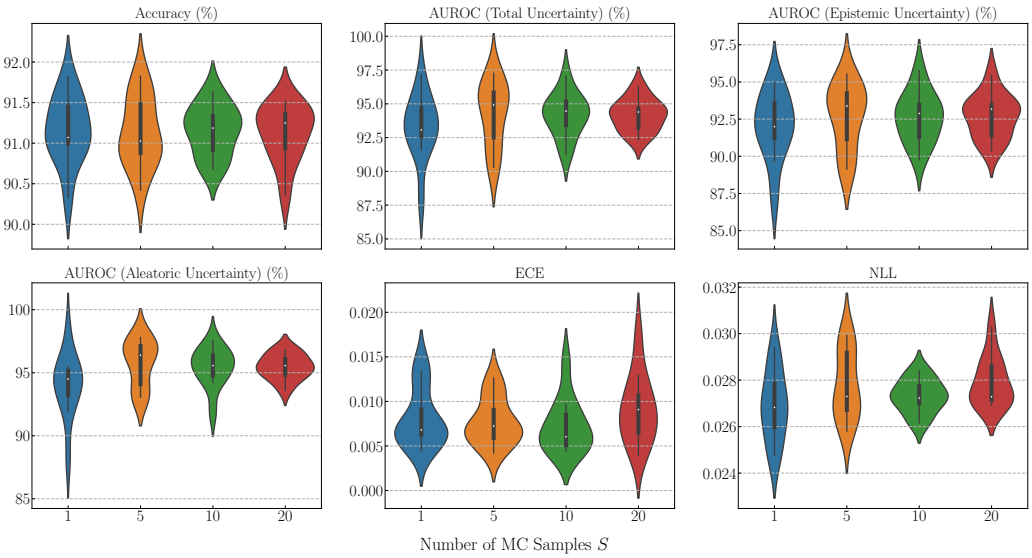
(a) FashionMNIST



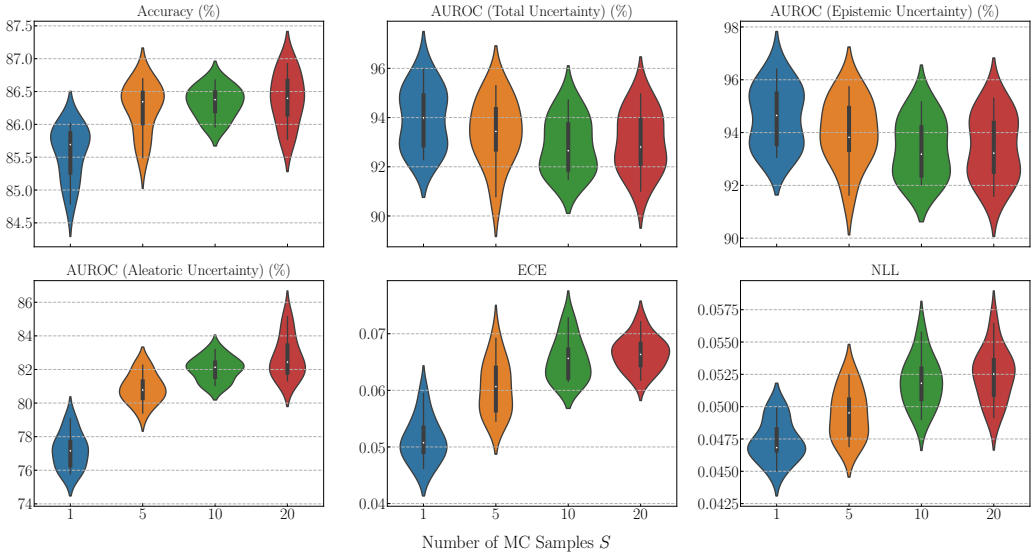
(b) CIFAR-10

Figure 23: Effect of the inducing input sampling method on in- and out-of-distribution evaluation metrics. The x-axis shows the fraction of the inducing inputs used for each gradient step that was sampled from the training batch. "0.0" means that no inducing inputs were sampled from the training data, and "1.0" means that all inducing inputs were sampled from the training set. For all experiments, 10 inducing inputs were sampled for each gradient step. A prior variance of 10 was used for FashionMNIST and a prior variance of 1.0 was used for CIFAR-10. 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.

F.5 Number of Monte Carlo Samples



(a) FashionMNIST



(b) CIFAR-10

Figure 24: Effect of the number of Monte Carlo samples used to evaluate the expected log-likelihood in the variational objective on in- and out-of-distribution evaluation metrics. The x-axis shows the number of Monte Carlo samples used. For all experiments, 10 inducing inputs were sampled for each gradient step. A prior variance of 10 was used for FashionMNIST and a prior variance of 1.0 was used for CIFAR-10. Out-of-distribution detection is performed on MNIST for models trained on FashionMNIST and on SVHN for models trained on CIFAR-10.