Function-Space Regularization in Neural Networks: A Probabilistic Perspective

Tim G. J. Rudner [1] **Sanyam Kapoor** [1] **Shikai Qiu** [1] **Andrew Gordon Wilson** [1]

Abstract

Parameter-space regularization in neural network optimization is a fundamental tool for improving generalization. However, standard parameter-space regularization methods make it challenging to encode explicit preferences about desired predictive *functions* into neural network training. In this work, we approach regularization in neural networks from a probabilistic perspective and show that by viewing parameter-space regularization as specifying an empirical prior distribution over the model parameters, we can derive a probabilistically well-motivated regularization technique that allows explicitly encoding information about desired predictive functions into neural network training. This method—which we refer to as function-space empirical Bayes (FS-EB)—includes both parameter- and function-space regularization, is mathematically simple, easy to implement, and incurs only minimal computational overhead compared to standard regularization techniques. We evaluate the utility of this regularization technique empirically and demonstrate that the proposed method leads to near-perfect semantic shift detection, highly-calibrated predictive uncertainty estimates, successful task adaption from pre-trained models, and improved generalization under covariate shift.

1. Introduction

The primary goal of machine learning is to find *functions* that represent relationships in data. Yet, most regularization methods in modern machine learning are expressed solely in terms of desired function parameters instead of the desired functions themselves.

[1]New York University, USA. Correspondence to: Tim G. J. Rudner <tim.rudner@nyu.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

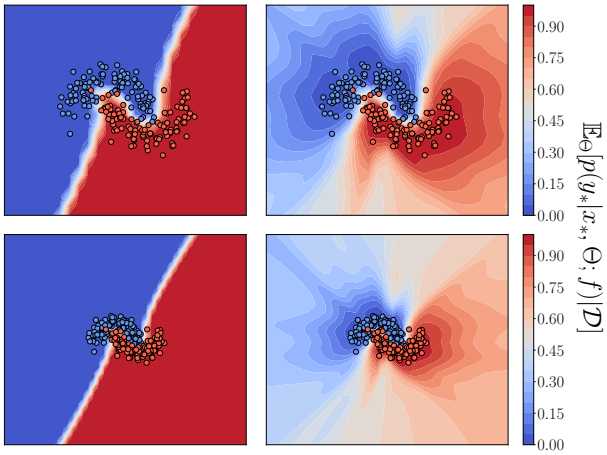


Figure 1: Predictive distributions obtained by training on the *Two Moons* datasets using standard parameter-space maximum a posteriori estimation (**Left**) and function-space empirical Bayes (FS-EB) (**Right**) in a two-layer MLP. FS-EB results in better-calibrated predictive uncertainty away from the training data, reflecting the inductive bias of the empirical prior distribution over the neural network parameters.

In this work, we propose a probabilistic inference method that results in an optimization objective that features both explicit parameter- and function-space regularization. To obtain such an optimization objective, we approach function-space regularization in deep neural networks from a probabilistic perspective and define an empirical prior distribution over parameters that allows explicitly encoding relevant prior information about the data-generating process into training. The resulting regularizer is mathematically simple, easy to implement, and effectively induces training dynamics that encourage solutions in parameter space that are consistent with both the encoded prior information about the network parameters and the desired functions. We refer to the probabilistic method as function-space empirical Bayes (FS-EB).

To derive an optimization objective that explicitly features parameter- and function-space regularization, we consider an empirical Bayes framework and specify an empirical prior distribution that reflects our prior beliefs about the

model parameter and the predictive function induced by them. More specifically, we consider a two-part inference problem: (i) an auxiliary inference problem for finding a posterior that can be used as an empirical prior and (ii) a primary inference problem, where we use the empirical prior and an observation model of the data to perform Bayesian inference.

To obtain an empirical prior that includes both parameter- and function-spaces regularizers, we consider an auxiliary inference problem, where the posterior distribution would reflect both prior beliefs about the neural network parameters (via a prior distribution over the parameters) as well as preferences about desired predictive functions (via a likelihood function that favors functions consistent with a specific distribution over functions).

We evaluate deterministic neural networks trained with the proposed regularized optimization objective on a broad range of standard classification, real-world domain adaption, and machine learning safety benchmarking tasks. We find that the proposed method successfully biases neural network training dynamics towards solutions that reflect the inductive biases of prior distributions over neural network functions, which can yield improved predictive performance and leads to significantly improved uncertainty quantification vis-à-vis standard parameter-space regularization and state-of-the-art function-space regularization methods.

To summarize, our key contributions are as follows:

- In Section 3.1, we specify an auxiliary inference problem, which allows us to obtain an analytically tractable unnormalized empirical prior distribution that reflects both prior beliefs about the neural network parameters and preferences about desired predictive functions.
- In Sections 3.2 and 3.3, we show how to perform tractable maximum a posteriori estimation and approximate posterior inference in neural networks using this unnormalized empirical prior and derive an optimization objective that features both parameter- and function-spaces regularization. We refer to this approach as function-space empirical Bayes (FS-EB).
- In Section 5, we present an empirical evaluation in which we compare highly-tuned parameter- and function-space regularization baselines to neural networks trained with FS-EB regularization and find that FS-EB yields (i) near-perfect semantic shift detection, (ii) highly-calibrated predictive uncertainty estimates, (iii) successful task adaption from pre-trained models, and (iv) improved generalization under covariate shift.

The code for our experiments can be accessed at:
`https://github.com/timrudner/function-space-empirical-bayes`.

2. Background

We will first review relevant background on probabilistic inference and related parameter-space and function-space regularization methods.

Consider supervised learning problems with N i.i.d. data realizations $\mathcal{D} = \{x^{(n)}, y^{(n)}\}_{n=1}^N = (\mathbf{x}_\mathcal{D}, \mathbf{y}_\mathcal{D})$ of inputs $x \in \mathcal{X}$ and targets $Y \in \mathcal{Y}$ with input space $\mathcal{X} \subseteq \mathbb{R}^D$ and target space $\mathcal{Y} \subseteq \mathbb{R}^K$ for regression and $\mathcal{Y} \subseteq \{0,1\}^K$ for classification tasks with K classes.

2.1. Parameter-Space Maximum A Posteriori Estimation

For supervised learning tasks, we define a parametric observation model $p_{Y|X,\Theta}(y \,|\, x, \theta; f)$ with mapping $f(\cdot\,; \theta) \doteq h(\cdot\,; \theta_h)\theta_L$ and a *prior* distribution over the parameters, $p_\Theta(\theta)$. Maximum a posteriori (MAP) estimation seeks to find the most likely setting θ^{MAP} of the quantity θ (under the probabilistic model) given the data. Since, by Bayes' Theorem, the implied posterior is proportional to the joint probability density given by the product of the *likelihood* of the parameters under the data $p_{Y|X,\Theta}(y_\mathcal{D} \,|\, x_\mathcal{D}, \theta)$ and the prior, that is,

$$p_{\Theta|Y,X}(\theta \,|\, y_\mathcal{D}, x_\mathcal{D}) \propto p_{Y|X,\Theta}(y_\mathcal{D} \,|\, x_\mathcal{D}, \theta) p_\Theta(\theta),$$

MAP estimation seeks to find the mode of the joint probability density $p(y_\mathcal{D} \,|\, x_\mathcal{D}, \theta)p(\theta)$ (Bishop, 2006; Murphy, 2013). Under a likelihood that factorizes across the data points given parameters θ,

$$p(y_\mathcal{D} \,|\, x_\mathcal{D}, \theta) \doteq \prod_{n=1}^N p(y_\mathcal{D}^{(n)} \,|\, x_\mathcal{D}^{(n)}, \theta), \tag{1}$$

the MAP optimization objective can be expressed as

$$\mathcal{L}^{\text{MAP}}(\theta) = \sum_{n=1}^N \log p_{Y|X,\Theta}(y_\mathcal{D}^{(n)} \,|\, x_\mathcal{D}^{(n)}, \theta) + \log p_\Theta(\theta).$$

The log-likelihood in the MAP optimization objective corresponds to a scaled negative mean squared error (MSE) loss function under a Gaussian likelihood (used for regression) and to a negative cross-entropy loss function under a categorical likelihood (used for classification).

The most common instantiations of parameter-space MAP estimation are L_1- and L_2-norm parameter regularization, which are also known as LASSO regression and weight decay or ridge regression, respectively. More specifically, choosing a prior $p(\theta) = \mathcal{N}(\theta; \mathbf{0}, \sigma_0^2 I)$ leads to the standard L_2-norm regularization (also known as weight decay) and $p(\theta) = \text{Laplace}(\theta; \mathbf{0}, bI)$ leads to the sparsity-inducing L_1-norm regularization (also known as LASSO) (Bishop, 2006; Murphy, 2013), making parameter-space MAP estimation one of the most widely used optimization frameworks in modern machine learning.

2.2. Function-Space Maximum A Posteriori Estimation

Wolpert (1993) considered posterior inference over functions evaluated at a finite set of context points, $\hat{x} \doteq \{x_1, ..., x_M\}$ to find the most likely parameters that represent the most likely function under the posterior distribution over functions.

Letting the set of input points $\hat{x}$ at which the function is evaluated contain the training data such that $x_{\mathcal{D}} \subseteq \hat{x}$, we can write the posterior distribution over functions at $\hat{x}$ as

$$p(f(\hat{x}) \mid y_{\mathcal{D}}, \hat{x}) = p(y_{\mathcal{D}} \mid x_{\mathcal{D}}, f(\hat{x}))p(f(\hat{x}) \mid \hat{x})/p(y_{\mathcal{D}} \mid \hat{x})$$

and express the mode of the posterior via the finite-point function-space MAP estimate $f(\hat{x}; \theta^{\text{FSMAP}})$ where θ^{FSMAP} is the mode of the finite-point function-space posterior:

$$\theta^{\text{FSMAP}} \doteq \arg\max_{\theta \in \mathbb{R}^P} p(y_{\mathcal{D}} \mid f(\hat{x}; \theta))p(f(\hat{x}; \theta) \mid \hat{x}).$$

To find the finite-points function-space MAP estimate, we need to be able to maximize the joint density

$$p(y_{\mathcal{D}} \mid f(\hat{x}; \theta))p(f(\hat{x}; \theta) \mid \hat{x})$$

with respect to θ. While the first term is the likelihood of the data given model parameters θ, the prior density $p(f(\hat{x}; \theta) \mid \hat{x})$ is not in general tractable. However, assuming that f is a neural network with a standard parameterization (e.g., a multi-layer perceptron) and the set of evaluation points is sufficiently large so that $MK \geq P$, using a generalization of the change-of-variables formula, Wolpert (1993) showed that the induced prior density is given by

$$p(f(\hat{x}\,; \theta)) = p(\theta) \det{}^{-1/2}(G(\theta)),$$

where $G(\theta)$ is a P-by-P matrix defined by

$$G(\theta) \doteq (\partial f(\hat{x}\,; \theta)/\partial\theta)^\top (\partial f(\hat{x}\,; \theta)/\partial\theta)$$

and $\partial f(\hat{x}\,; \theta)/\partial\theta$ is the MK-by-P Jacobian matrix of $f(\hat{x}\,; \theta)$ with respect to the parameters θ. To find θ^{FSMAP}, one can maximize the log-joint density function,

$$\begin{aligned}&\log p(f(\hat{x}; \theta) \mid y_{\mathcal{D}}, \hat{x}) \\ &= \log p(y_{\mathcal{D}} \mid f(\hat{x}; \theta)) + \log p(\theta) - \frac{1}{2} \log\det(G(\theta)).\end{aligned}$$

That is, function-space MAP estimation results in an optimization objective that includes parameter- and function-space regularization. Unfortunately, computing the correction term is analytically intractable and computationally infeasible for large neural networks. Motivated by function-space MAP estimation, in Section 3.2, we present an alternative probabilistic model that also features both parameter- and function-space regularization but is analytically tractable and scalable to large neural networks.

2.3. Function-Space Variational Inference

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 $\Theta \in \mathbb{R}^P$ and letting the function mapping defined by a neural network architecture be given by $f : \mathcal{X} \times \mathbb{R}^P \to \mathbb{R}^K$, then $f(\cdot\,; \Theta)$ is a random function. For a parameter realization θ, we obtain a function realization, $f(\cdot\,; \theta)$, and when evaluated at a finite collection of points $\hat{x} \doteq \{x_1, ..., x_M\}$, $f(\hat{x}; \Theta)$ is a multivariate random variable.

Instead of seeking to infer a posterior distribution over parameters, we may equivalently frame Bayesian inference in stochastic neural networks as inferring a posterior distribution over functions (Sun et al., 2019b; Rudner et al., 2022a). Given a prior distribution over parameters $p(\theta)$, the probability density of the corresponding induced prior distribution over functions $p(f(\cdot))$ evaluated at a finite set of evaluation points x, can be expressed as

$$p_{F(x)}(f(x)) = \int_{\mathbb{R}^P} p_\Theta(\theta')\, \delta(f(x; \theta) - f(x; \theta'))\, \mathrm{d}\theta',$$

where $\delta(\cdot)$ is the Dirac delta function. The probability density of the posterior distribution over functions $p(f(\cdot)|\mathcal{D})$ induced by the posterior distribution over parameters $p(\theta|\mathcal{D})$, evaluated at a finite set of points, can be defined analogously and is given by

$$\begin{aligned}&p_{F(x)|\mathcal{D}}(f(x) \mid \mathcal{D}) \\ &= \int_{\mathbb{R}^P} p_{\Theta|\mathcal{D}}(\theta' \mid \mathcal{D})\, \delta(f(x\,; \theta) - f(x\,; \theta'))\, \mathrm{d}\theta'.\end{aligned}$$

Finally, defining a variational distribution over functions $q(F(\cdot))$ induced by a variational distribution over parameters $q(\theta)$, we can frame inference over

$$q_{F(x)}(f(x)) = \int_{\mathbb{R}^P} q_\Theta(\theta')\, \delta(f(x; \theta) - f(x; \theta'))\, \mathrm{d}\theta',$$

we can frame posterior inference over stochastic functions $F(\cdot)$ variationally as

$$\min_{q_\Theta \in \mathcal{Q}} \mathbb{D}_{\text{KL}}(q_{F(\cdot)} \,\|\, p_{F(\cdot)|\mathcal{D}}),$$

where $\mathcal{Q}$ is a variational family. Equivalently, we can express the inference problem as

$$\max_{q_\Theta \in \mathcal{Q}} \mathbb{E}_{q_{F(\cdot)}}[\log p(y_{\mathcal{D}} \mid x_{\mathcal{D}}, F(\cdot))] - \mathbb{D}_{\text{KL}}(q_{F(\cdot)} \,\|\, p_{F(\cdot)}),$$

where $\mathbb{D}_{\text{KL}}(q_{F(\cdot)} \,\|\, p_{F(\cdot)})$ is an explicit regularizer on the variational distribution over functions $q(F(\cdot))$. Rudner et al. (2022a), Sun et al. (2019b), and Ma & Hernández-Lobato (2021) have proposed tractable approximations to this objective. The function-space variational inference (FS-VI) approach by Rudner et al. (2022a) is a state-of-the-art approximate inference method for BNNs.

3. Function-Space Empirical Bayes

Instead of considering standard, uninformative prior distributions over parameters, we consider an empirical prior distribution over parameters, which allows us to obtain an optimization objective that combines the benefits of both standard parameter-space and explicit function-space regularization. To obtain such an objective, we will consider a two-part inference procedure. First, we will consider an auxiliary inference problem to derive an analytically tractable unnormalized empirical prior distribution. We will then show how to incorporate this empirical prior into MAP estimation and variational inference for the neural network parameters. The resulting optimization objectives feature both explicit parameter- and function-space regularization.

3.1. Empirical Priors via Distributions over Functions

We begin by specifying the auxiliary inference problem. Let $\hat{x} = \{x_1, ..., x_M\}$, be a set of context points with corresponding labels $\hat{y}$, and define a corresponding likelihood function $\hat{p}_{Y|X,\Theta}(\hat{y} \,|\, \hat{x}, \theta; f)$ and a prior over the model parameters, $p_\Theta(\theta)$. For notational simplicity, we will drop the subscripts going forward except when needed for clarity. By Bayes' Theorem, the posterior under the context points and labels is given by

$$\hat{p}(\theta \,|\, \hat{y}, \hat{x}) \propto \hat{p}(\hat{y} \,|\, \hat{x}, \theta; f) p(\theta). \tag{2}$$

To define a likelihood function that induces a posterior with desirable properties, we consider the following stochastic linear model for an arbitrary set of points $x \doteq \{x_1, ..., x_{M'}\}$,

$$Z_k(x) \doteq h(x; \phi_0)\Psi_k + \varepsilon$$

$$\text{with} \quad \Psi_k \sim \mathcal{N}(\psi; \mu, \tau_f^{-1} I) \quad \text{and} \quad \varepsilon \sim \mathcal{N}(\mathbf{0}, \tau_f^{-1} I),$$

for output dimensions $k = 1, ..., K$, where $h(\cdot\,; \phi_0)$ is the feature mapping used to define f evaluated at a set of fixed feature parameters ϕ_0, μ is a set of mean parameters, and τ_f is a precision parameter. This stochastic linear model induces a distribution over functions, which—when evaluated at $\hat{x}$—is given by

$$\mathcal{N}(z_k(\hat{x}); h(\hat{x}; \phi_0)\mu_k, \tau_f K(\hat{x}, \hat{x}; \phi_0)),$$

where

$$K(\hat{x}, \hat{x}; \phi_0) \doteq h(\hat{x}; \phi_0) h(\hat{x}; \phi_0)^\top + I \tag{3}$$

is an M-by-M covariance matrix. Letting $\mu = \mathbf{0}$, we obtain

$$p(z_k \,|\, \hat{x}) = \mathcal{N}(z_k; \mathbf{0}, \tau_f K(\hat{x}, \hat{x}; \phi_0)).$$

Viewing this probability density over function evaluations as a likelihood function parameterized by θ, we define

$$\hat{p}(\hat{y}_k \,|\, \hat{x}, \theta; f) \doteq \mathcal{N}(\hat{y}_k; f(\hat{x}; \theta)_k, \tau_f K(\hat{x}, \hat{x}; \phi_0)), \tag{4}$$

with labels $\hat{y} \doteq \{\mathbf{0}, ..., \mathbf{0}\}$. This likelihood function favors parameters θ for which $f(\hat{x}; \theta)$ has high likelihood under the induced prior distribution over functions in Equation (4). Letting the likelihood factorize across output dimensions,

$$\hat{p}(\hat{y} \,|\, \hat{x}, \theta; f) \doteq \prod_{k=1}^{K} \hat{p}(\hat{y}_k \,|\, \hat{x}, \theta; f),$$

defining the prior distribution over parameters as $p(\theta) = \mathcal{N}(\theta; \mathbf{0}, \tau_\theta^{-1})$, and taking the log of the analytically tractable joint density $\hat{p}(\hat{y} \,|\, \hat{x}, \theta; f)p(\theta)$, we obtain

$$\log \hat{p}(\hat{y} \,|\, \hat{x}, \theta; f) + \log p(\theta)$$
$$\propto -\sum_{k=1}^{K} \frac{\tau_f}{2} f(\hat{x}; \theta)_k^\top K(\hat{x}, \hat{x}; \phi_0)^{-1} f(\hat{x}; \theta)_k - \frac{\tau_\theta}{2} \|\theta\|_2^2,$$

with proportionality up to an additive constant independent of θ. Defining

$$\mathcal{J}(\theta, \hat{x}) \doteq -\sum_{k=1}^{K} \frac{\tau_f}{2} d_M(f(\hat{x}; \theta)_k, K(\hat{x}, \hat{x}; \phi_0)) - \frac{\tau_\theta}{2} \|\theta\|_2^2, \tag{5}$$

where $d_M(v, K) \doteq v^\top K^{-1} v$ is the Mahalanobis distance between v and $\mathbf{0}$. We therefore obtain

$$\arg\max_\theta \hat{p}(\theta \,|\, \hat{y}, \hat{x}) = \arg\max_\theta \mathcal{J}(\theta, \hat{x}).$$

and hence, maximizing $\mathcal{J}(\theta, \hat{x})$ with respect to θ is mathematically equivalent to maximizing the posterior $\hat{p}(\theta \,|\, \hat{y}, \hat{x})$ and leads to functions that are likely under the distribution over functions induced by the neural network mapping while being consistent with the prior over the network paramters.

3.2. Empirical Bayes Maximum A Posteriori Estimation

We can now move on to the main inference problem. Using the training data $\mathcal{D}$, we wish to find a predictive function that fits the training data, generalizes well, and has well-calibrated predictive uncertainty. To obtain such a predictive function, we will perform MAP estimation using the posterior $\hat{p}(\theta \,|\, \hat{y}, \hat{x})$ as an empirical prior over parameters.

Since the posterior considered above is proportional to an analytically tractable joint distribution, performing MAP estimation using the posterior from the secondary inference problem as an empirical prior is straightforward. Defining a probabilistic model with the empirical prior,

$$p(\theta \,|\, y_\mathcal{D}, x_\mathcal{D}) \propto p(y_\mathcal{D} \,|\, x_\mathcal{D}, \theta) \hat{p}(\theta \,|\, \hat{y}, \hat{x}), \tag{6}$$

we can perform MAP estimation by maximizing the empirical-MAP optimization objective,

$$\log p(\theta \,|\, y_\mathcal{D}, x_\mathcal{D}) \propto \log p(y_\mathcal{D} \,|\, x_\mathcal{D}, \theta) + \log \hat{p}(\theta \,|\, \hat{y}, \hat{x}),$$

which is analytically tractable and can be expressed as

$$\mathcal{L}^{\text{EB-MAP}}(\theta) \doteq \sum_{n=1}^{N} \log p(y_\mathcal{D}^{(n)} \,|\, x_\mathcal{D}^{(n)}, \theta) + \mathcal{J}(\theta, \hat{x}). \tag{7}$$

This objective contains explicit penalties on both the parameter values (via the parameter norm $\|\theta\|_2^2$) as well as the induced function values on the set of context points (via the Mahalanobis distance between function evaluations and the zero vector, $d_M(f(\hat{x};\theta)_k, K(\hat{x},\hat{x};\phi_0))$).

3.3. Empirical Bayes Variational Inference

While the regularizer in Equation (5) may induce the desired behavior for a *given* set of context points $\hat{x}$, we may instead wish to specify a *distribution over context points* to cover a larger region of input space. To obtain a tractable objective function for this setting, we consider a variational formulation of the inference problem. Slightly changing the notation (using θ' instead of θ), the probabilistic model in which we wish to perform inference—defined in terms of both the empirical prior and a prior distribution over the set of context points—is given by

$$p(\theta', \hat{x} \mid y_{\mathcal{D}}, x_{\mathcal{D}}) \propto p(y_{\mathcal{D}} \mid x_{\mathcal{D}}, \theta')\hat{p}(\theta' \mid \hat{y}, \hat{x})p(\hat{x}), \quad (8)$$

with the empirical prior

$$\hat{p}(\theta' \mid \hat{y}, \hat{x}) \propto \hat{p}(\hat{y} \mid \hat{x}, \theta'; f)p(\theta'). \quad (9)$$

Now, defining a variational distribution

$$q(\theta', \hat{x}) \doteq q(\theta')q(\hat{x}),$$

we can frame the inference problem of finding the posterior $p(\theta', \hat{x} \mid y_{\mathcal{D}}, x_{\mathcal{D}})$ as a problem of optimization,

$$\min_{q_{\Theta',\hat{X}} \in \mathcal{Q}} D_{\mathrm{KL}}(q_{\Theta',\hat{X}} \parallel p_{\Theta',\hat{X} \mid Y_{\mathcal{D}}, X_{\mathcal{D}}}),$$

where $\mathcal{Q}$ is a variational family. If $q_{\Theta',\hat{X}} \in \mathcal{Q}$, then the solution to the variational minimization problem is equal to the exact posterior. Defining $q(\hat{x}) \doteq p(\hat{x})$, which further constrains the variational family, the optimization problem simplifies to

$$\min_{q_{\Theta'} \in \mathcal{Q}} \mathbb{E}_{p_{\hat{X}}} \left[D_{\mathrm{KL}}(q_{\Theta'} \parallel p_{\Theta' \mid Y_{\mathcal{D}}, X_{\mathcal{D}}})\right],$$

which can equivalently be expressed as maximizing the variational objective

$$\mathbb{E}_{q_{\Theta'}}[\log p(y_{\mathcal{D}} \mid x_{\mathcal{D}}, \Theta'; f)] - \mathbb{E}_{p_{\hat{X}}}[D_{\mathrm{KL}}(q_{\Theta'} \parallel p_{\Theta' \mid \hat{Y}, \hat{X}})].$$

To obtain a tractable estimator of the regularization term, we first note that we can write

$$\begin{aligned}&\mathbb{E}_{p_{\hat{X}}}[D_{\mathrm{KL}}(q_{\Theta'} \parallel p_{\Theta' \mid \hat{Y}, \hat{X}})]] \\ &= \mathbb{E}_{p_{\hat{X}}}[\mathbb{E}_{q_{\Theta'}}[\log q(\Theta')] - \mathbb{E}_{q_{\Theta'}}[\log p(\Theta' \mid \hat{Y}, \hat{X})]],\end{aligned}$$

where the first term is the negative entropy and the second term is the negative cross-entropy. Defining a mean-field variational distribution $q(\theta') \doteq \mathcal{N}(\theta'; \theta, \sigma^2 I)$ with learnable θ and very small and fixed σ^2 (e.g., $\sigma^2 = 10^{-20}$), the negative entropy term will be constant in θ, and letting $p(\theta) = \mathcal{N}(\theta; \mathbf{0}, \tau_\theta^{-1})$ as before, we get

$$\begin{aligned}&\mathbb{E}_{p_{\hat{X}}}[\mathbb{E}_{q_{\Theta'}}[\log p(\Theta' \mid \hat{Y}, \hat{X})]] \\ &\propto \mathbb{E}_{p_{\hat{X}}}\left[\mathbb{E}_{q_{\Theta'}}\left[\log \hat{p}(\hat{Y} \mid \hat{X}, \Theta'; f)\right] + \mathbb{E}_{q_{\Theta'}}\left[-\frac{\tau_\theta}{2}\|\Theta'\|_2^2\right]\right],\end{aligned}$$

up to an additive constant independent of θ. From this expression, we can obtain an unbiased estimator of the KL divergence using simple Monte Carlo estimation:

$$\mathcal{F}(\theta) \doteq -\frac{1}{IJ}\sum_{i=1}^{I}\sum_{j=1}^{J} \mathcal{J}(\theta + \sigma\epsilon^{(j)}, \hat{X}^{(i)}) + C \quad (10)$$
$$\text{with} \quad \hat{X}^{(i)} \sim p_{\hat{X}} \quad \text{and} \quad \epsilon^{(j)} \sim \mathcal{N}(\mathbf{0}, I)$$

for $i = 1, ..., I$, $j = 1, ..., J$, and an additive constant C independent of θ. This regularizer is an estimator of the expectation of $\mathcal{J}(\Theta, \hat{X})$ under $q_{\Theta'}$ and $p_{\hat{X}}$. Finally, we obtain the variational objective

$$\mathcal{L}^{\text{EB-VI}}(\theta) = \frac{1}{S}\sum_{n=1}^{N}\sum_{s=1}^{S} \log p(y_{\mathcal{D}}^{(n)} \mid x_{\mathcal{D}}^{(n)}, \theta + \sigma\epsilon^{(s)}) + \mathcal{F}(\theta), \quad (11)$$

with $\epsilon^{(s)} \sim \mathcal{N}(\mathbf{0}, I)$. This objective factorizes across training data points and, as such, is amenable to stochastic gradient descent. This objective is used in the empirical evaluation in Section 5. We will refer to this method as function-space empirical Bayes (FS-EB).

3.4. Function-Space Regularization via Empirical Priors

The tractable empirical-Bayes MAP estimation and variational inference objectives in Equations (7) and (11), respectively, are both defined in terms of the empirical-Bayes regularizer $J(\theta, \hat{x})$ given in Equation (5).

First, unlike function-space regularizers proposed in prior work (e.g., Bietti et al., 2019; Benjamin et al., 2018; Sun et al., 2019b; Rudner et al., 2022a;b; Chen et al., 2022), the regularizer $\mathcal{J}(\theta, \hat{x})$, explicitly features parameter-space regularization. Prior distributions over parameters, such as isotropic Gaussians or the Laplace distribution, are well-established and have been demonstrated to yield parameter MAP estimates that define predictive functions that generalize well. Second, via the labels $\hat{y} = \{\mathbf{0}, ..., \mathbf{0}\}$ used in the likelihood function, the parameters θ are encouraged to be concentrated around values that fit the training data and are consistent with both the prior distribution over parameters—which favors parameters θ with small norm $\|\theta\|_2^2$—and the likelihood function—which favors parameters θ that produce zero predictions, corresponding to high-entropy predictive distributions in classification settings and a reversion to the data mean in regression settings with normalized data. Third, for non-singleton sets of context points $\hat{x}$, the likelihood function enforces a smoothness constraint via its

covariance matrix and encourages parameters that induce functions that have high likelihood under the induced distribution over functions defined Equation (4)—which has been shown introduce desirable inductive biases into the learned model (Wilson & Izmailov, 2020; Rudner et al., 2022a;b).

3.5. Specifying Distributions over Sets of Context Points

Careful specification of $p_{\hat{X}}$ is crucial for ensuring that the empirical-Bayes regularizer effectively encourages desired properties in the learned predictive functions. A simple approach to specifying $p_{\hat{X}}$ is to define the context distribution as an empirical distribution given by a dataset that is meaningfully related to the training data. For example, we may choose an unaltered subset of the training data, corruptions/augmentations of the training data (using standard augmentations such as cropping, blurring, pixelation, etc.), or a related dataset, such as KMNIST when training on FashionMNIST or CIFAR-100 when training on CIFAR-10, as the context distribution. In principle, the more the most relevant regions of a given problem-specific input space (e.g., the space of natural images for general image classification) are covered by a context distribution $p_{\hat{X}}$, the more likely the learned function will be drawn towards the prior distribution over functions evaluated at these parts of input space.

3.6. Specifying Prior Distributions over Functions

When a pretrained model is available, a likelihood $\hat{p}(\hat{y} \mid \hat{x}, \theta; f)$ can be constructed from a prior distribution over functions by specifying ϕ_0 in $h(\hat{x}; \phi_0)$ to be the pretrained model parameters. If a pretrained model is unavailable, ϕ_0 can be specified by randomly initializing the network parameters using any standard initialization scheme, which also induces desirable inductive biases (Wilson & Izmailov, 2020).

4. Related Work

Krogh & Hertz (1991) argued that explicit regularization via weight decay, that is, an L_2-norm penalty on the parameters, can significantly improve generalization. This approach is now standard practice for training parametric models, including large neural networks. Weight decay corresponds to maximum a posteriori estimation in probabilistic models with a Gaussian prior distribution over the model parameters. Joo & Chung (2020) further demonstrated the effectiveness of explicit regularization for calibration of neural networks. Our work takes this case further by regularizing directly in the function space.

Wolpert (1993) argued that the true goal of maximum a posteriori estimation in parametric models—and, as such, of parameter-space regularization—is to find the most likely function mapping that describes the given data and the prior while the parameter-space representation of the network is only a means to an end. However, in non-linear parametric models, since maximum a posteriori estimation is not invariant under parameterization, the function implied by the most likely parameters can differ significantly from the most probable function (Denker & LeCun, 1990). Using the generalized change-of-variables formula for probability distributions to get the implied distribution over functions from the distribution over parameters, Wolpert (1993) introduced a correction term to standard parameter-space regularization with weight decay limited to small neural networks. In contrast, we provide an alternative model formulation that leads to tractable function-space regularization for any neural network architecture.

Wang et al. (2019) reasoned why a good approximation to the parameter-space posterior does not necessarily correspond to better predictive performance because of symmetries in overparameterized neural networks. Empirically, Joo & Chung (2020) provided evidence that L_p norm regularization in function space improves generalization in neural network models while also improving calibration. Bietti et al. (2019) proposed to use the Jacobian norm as a lower bound on the function norm and Bietti & Mairal (2018) constructed an RKHS which contains CNN prediction functions. Chen et al. (2022) use a Mahalanobis distance regularizer between logits, with the covariance matrix given by the empirical neural tangent kernel. In this work, we instead take an empirical Bayes approach to derive a function-space regularization objective from inference in a probabilistic model of the data-generating process.

In the context of approximate Bayesian inference, Sun et al. (2019a) proposed to minimize the divergence between two distributions over functions via a function-space evidence lower bound (ELBO), but Burt et al. (2020) showed that the inference problem as considered in Sun et al. (2019b) is not well-defined for neural network variational distributions with Gaussian process priors. Other approaches to approximate function-space inference have been proposed (Ma et al., 2018; Ober & Aitchison, 2020; Ma & Hernández-Lobato, 2021). By instead linearizing the function mapping to obtain a tractable distribution over functions, Rudner et al. (2022a) introduced an effective and scalable approximation to make function-space variational inference effective and scalable to deep neural networks. Titsias et al. (2019) applied functional regularization using Gaussian process priors to handle catastrophic forgetting in continual learning and Rudner et al. (2022b) use function-space variational inference to prevent catastrophic forgetting by encouraging neural networks to match an empirical prior distribution over functions. We reiterate, however, that our work does not aim to propose a new approximate Bayesian inference approach. Instead, we investigate the utility of approximate inference with a function-space regularizer specified via empirical Bayes on the parameters.

5. Empirical Evaluation

In this section, we evaluate empirical variational inference (FS-EB) along various dimensions—generalization (accuracy), uncertainty quantification (selective prediction, calibration), robustness (semantic shift detection, generalization under covariate shift), and transfer learning.

Overview. We assess whether FS-EB can improve the reliability of neural networks. We put a special emphasis on benchmarking tasks and evaluation metrics that assess reliability as a function of predictive accuracy and predictive uncertainty estimates. Across all benchmarking tasks, we find that FS-EB results in improved predictive uncertainty, evaluated in terms of log-likelihood, expected calibration error (ECE), and selective prediction when compared to standard parameter-space MAP (denoted by PS-MAP). Notably, we achieve *near-perfect* semantic shift detection on both CIFAR-10 and FashionMNIST against samples from datasets that were unseen during training and do not belong to the same distribution. We further demonstrate that FS-EB can often improve robustness to corruptions compared to parameter-space inference.

Illustrative Example. In Figure 1, we illustrate the effect of FS-EB on the *Two Moons* classification dataset. On one hand, a standard data fit using standard parameter-space MAP estimation shows that the model learns a decision boundary which roughly splits the space into two regions within which the model makes predictions with very high confidence. FS-EB, on the other hand, exhibits an increase in predictive uncertainty in regions further away from the training data, where it encourages the neural network to match the prior distribution over functions (via the empirical prior), providing a more reliable solution that aligns with our a priori desire of lower confidence predictions in regions of input space far away from the training data.

Setup. All of our methods are trained using a ResNet-18 architecture (He et al., 2016) with momentum SGD. All results are reported with mean and standard error over five trials. See Appendix A for details about hyperparameters.

Implementation. The optimization objective in Equation (11) can be implemented on top of standard training routines. It only requires the neural network feature $h(\hat{x}; \phi_0)$ and the predictions $f(\hat{x}; \theta)$ for a given sample of context points $\hat{x}$. In practice, we use only a single Monte Carlo sample per gradient step, that is, $I = J = 1$.

5.1. Selective Prediction

Selective prediction modifies the standard prediction pipeline by introducing a "reject option", $\perp$, via a gating mechanism defined by a selection function $s : \mathcal{X} \to \mathbb{R}$ that determines whether a prediction should be made for a given input point $x \in \mathcal{X}$ (El-Yaniv & Wiener, 2010; Rabanser et al., 2022). For a rejection threshold τ, the prediction model is then given by

$$(p(y \mid \cdot, \boldsymbol{\theta}; f), s)(x) = \begin{cases} p(y \mid x, \boldsymbol{\theta}; f) & s \leq \tau \\ \perp & \text{otherwise.} \end{cases} \tag{12}$$

To evaluate the predictive performance of a prediction model $(p(y \mid \cdot, \boldsymbol{\theta}; f), s)(x)$, we compute the predictive performance of the classifier $p(y \mid x, \boldsymbol{\theta}; f)$ over a range of thresholds τ, and summarize as the area under the selective prediction accuracy curve. Successful selective prediction models obtain high cumulative accuracy over many thresholds and can be applied in safety-critical real-world tasks where uncertainty-aware predictive accuracy is especially important.

Figure 2 shows that FS-EB can often provide better out-of-the-box for certain standard image corruptions, tested on the Corrupted CIFAR-10 (Hendrycks & Dietterich, 2019) dataset. We plot the selective prediction accuracy curves, that is, accuracy versus confidence, such that below a chosen confidence level τ, the sample is not being classified. Additionally, in Tables 1 and 2, we see that FS-EB improves the area under selective prediction curves, while improving the generalization of the classifier as measured by accuracy. In practice, a fraction $1 - \tau$ of the samples could get referred to a human expert for manual review. The area under the selective prediction accuracy curves, therefore, provides information about the reliability of a classifier.

5.2. Calibrated Predictive Uncertainty

As shown in Figure 1, PS-MAP tends to be very confident even far away from data. Such predictive behavior may often be undesirable. The expected calibration error (ECE; Naeini et al. (2015)) computes the alignment between accuracy and prediction of a classifier. In line with our illustration, through our benchmark experiments, we provide evidence that FS-EB is able to significantly improve classification calibration.

Following Naeini et al. (2015), an empirical ECE estimator is constructed by binning the maximum output probability of each sample into m bins $B_j \; \forall \, j \in [1, \ldots, m]$, such that

$$\widehat{\mathrm{ECE}} = \sum_{i=1}^{n} \frac{B_i}{n} \left| \mathrm{Accuracy}(B_i) - \mathrm{Confidence}(B_i) \right|, \tag{13}$$

where $\mathrm{Acc.}$ is the accuracy of each sample within each bin B_i, and $\mathrm{Conf.}$ is the mean of all maximum probability outputs of a classifier for each sample within the bin B_j. Therefore, a perfectly calibrated model has an ECE of zero, implying perfect alignment between the accuracy of the classifier and its confidence in the predictions.

In Tables 1 and 2, we verify that FS-EB significantly improves calibration while improving the generalization of the classifier as measured by accuracy.

Table 1: We report the accuracy (ACC.), negative log-likelihood (NLL), expected calibration error (ECE), and area under the selective prediction accuracy curve (SEL. PRED.) for FashionMNIST (Xiao et al., 2017) and FS-EB improves performance while improving calibration. $x_{\mathrm{C}} = \mathrm{KMNIST}$. Means and standard errors are computed over five seeds.

METHOD	ACC. ↑	SEL. PRED. ↑	NLL ↓	ECE ↓
PS-MAP	93.8%±0.0	**98.9%**±0.0	0.26±0.00	3.6%±0.0
FS-EB	**94.1%**±0.1	98.8%±0.0	**0.19**±0.00	**1.8%**±0.1
FS-VI	**94.1%**±0.0	98.4%±0.0	0.24±0.00	2.6%±0.1

Table 2: We report the accuracy (ACC.), negative log-likelihood (NLL), expected calibration error (ECE), and area under the selective prediction accuracy curve (SEL. PRED.) for CIFAR-10 (Krizhevsky, 2010) and FS-EB improves predictive performance and calibration. $x_{\mathrm{C}} = \mathrm{CIFAR\text{-}100}$. Means and standard errors are computed over five seeds.

METHOD	ACC. ↑	SEL. PRED. ↑	NLL ↓	ECE ↓
PS-MAP	94.9%±0.2	99.3%±0.0	0.21±0.01	3.0%±0.1
FS-EB	**95.1%**±0.1	**99.4%**±0.0	**0.20**±0.00	**2.1%**±0.1
FS-VI	92.9%±0.1	98.0%±0.0	0.31±0.00	4.0%±0.1

(a) Corruption Level 3

(b) Corruption Level 5

Figure 2: For a selected subset of corruptions as constructed by Corrupted CIFAR-10 (Hendrycks & Dietterich, 2019), we show the selective prediction curves for **(a)** corruption level 3 and **(b)** corruption level 5. A higher curve indicates better calibration for "reject option" in classification (El-Yaniv & Wiener, 2010). We find that FS-EB is often better out-of-the-box for certain standard image blurring and noise corruptions, indicating better calibration when compared to standard PS-MAP.

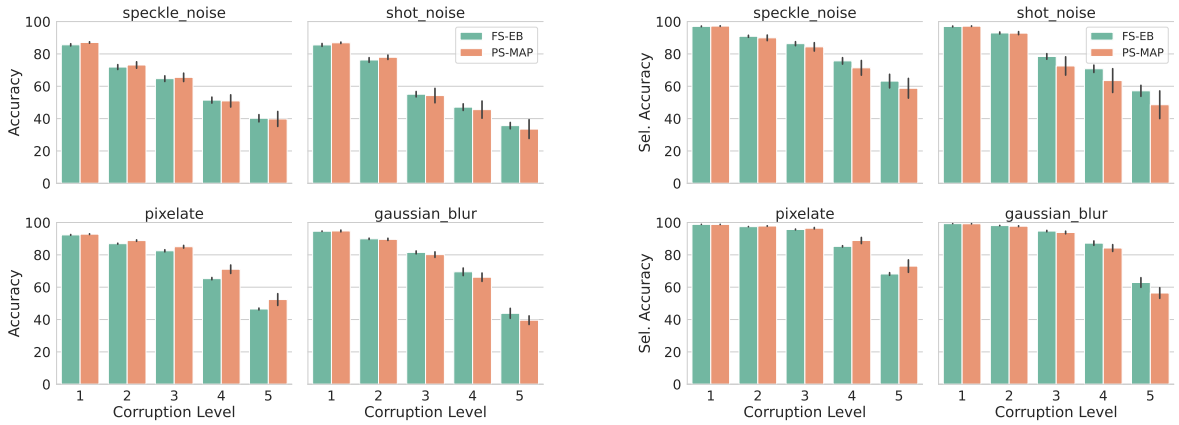
5.3. Highly-Accurate Semantic Shift Detection

So far, we have demonstrated that FS-EB can improve the quality of neural networks' predictive uncertainty on in-domain data. Another hallmark of a reliable model is its ability to detect semantic shifts in the data (Band et al., 2021; Nado et al., 2021). We assess whether the FS-EB generates predictive uncertainty estimates that enable successful semantic shift detection, that is, detection of input points whose true labels are semantically different from the training labels, and find that FS-EB can achieve *near-perfect* semantic shift detection in two image classification tasks. To simulate semantic shift, we present a classifier trained on FashionMNIST (Xiao et al., 2017), a grayscale collection of fashion items to distinguish against KMNIST (Clanuwat et al., 2018), with a dataset of handwritten Kuzushiji digits.

Using the predictive entropy of the classifier for each input sample from both FashionMNIST and KMNIST, we build another binary classifier to detect semantic shifts using simply the threshold of predictive entropy. We are able to detect semantic shift with *near-perfect* accuracy of 99.9%. We come to a similar conclusion when detecting semantic shift between CIFAR-10 (Krizhevsky, 2010), a collection of tiny images of objects and SVHN (Netzer et al., 2011), a collection of street view house numbers. Numerical results are summarized in Table 3.

Table 3: We compute the area under the ROC of a classifier using the predictive entropy on the in-distribution samples and out-of-distribution samples x_{OOD} with semantic shift. For FashionMNIST, we use $x_{\mathrm{OOD}} = \mathrm{MNIST}$; for CIFAR-10, we use $x_{\mathrm{OOD}} = \mathrm{SVHN}$.

DATASET	METHOD	OOD AUROC ↑
FMNIST	PS-MAP	94.9%±0.4
	FS-EB (x_C = KMNIST)	**99.9%**±0.0
	FS-VI	98.0%±0.4
CIFAR-10	PS-MAP	93.0%±0.4
	FS-EB (x_C = CIFAR100)	**99.4%**±0.1
	FS-VI	99.0%±0.1



(a) Accuracy

(b) Selective Prediction AUC

Figure 3: For a randomly selected subset of corruptions as constructed by Corrupted CIFAR-10 (Hendrycks & Dietterich, 2019), we show that **(a)** FS-EB and PS-MAP achieve similar predictive accuracy, but **(b)** FS-EB leads to better selective prediction (as measured by the area under the selective prediction accuracy curve). The improvement in selective prediction indicates that FS-EB produces more accurate uncertainty estimates and is thus able to use the "reject option" more effectively, leading to more reliable classification. See Figures 4 and 5 in Appendix A for results on other common corruptions.

5.4. Generalization under Covariate Shift

Another essential property of a reliable classifier is graceful degradation under covariate shift. We assess the performance of FS-EB in terms of generalization under covariate shift. Using the CIFAR-10 Corrupted dataset (Hendrycks & Dietterich, 2019) at five different corruption intensity levels, we find that FS-EB can still generalize well. In Figure 3, we find that FS-EB often works out-of-the-box for generalization under common visual corruptions.

5.5. Improved Transfer Learning

In addition to training from scratch, we also investigate the utility of FS-EB for transfer learning, a paradigm that is now very common with the advent of large pretrained neural network models (Brown et al., 2020; Radford et al., 2021; Tran et al., 2022; Touvron et al., 2023).

We find that FS-EB improves uncertainty quantification of transfer-learned models without compromising predictive performance. Table 4 shows that FS-EB and PS-MAP reach the same level of accuracy and selective prediction AUC, but FS-EB significantly improves NLL, calibration as measured by ECE, and effective semantic shift detection, using a ResNet-18 (He et al., 2016) pretrained on ImageNet (Russakovsky et al., 2014).

In addition, we evaluate transfer-learned classifiers with FS-EB on real-world datasets. Using a ResNet-50 pretrained on ImageNet, we train classifiers on blindness detection, leaf disease classification, and melanoma detection and find that FS-EB often outperforms PS-MAP in generalization while significantly improving uncertainty quantification. These results are presented in Appendix A.8.

Table 4: Starting from a pretrained checkpoint of ResNet18 on ImageNet (Russakovsky et al., 2014), we report the performance on CIFAR-10 (Recht et al., 2018). FS-EB benefits predictive performance and calibration. Means and standard errors are computed over five seeds.

METHOD	ACC. ↑	SEL. PRED. ↑	NLL ↓	ECE ↓	OOD ↑
PS-MAP	96.2%±0.1	99.6%±0.0	0.13±0.01	3.2%±0.2	96.3%±0.7
FS-EB	96.2%±0.1	99.6%±0.0	**0.11**±0.00	**1.3%**±0.1	**98.9%**±0.1

6. Conclusion

We presented a probabilistic perspective on function-space regularization in neural networks and used it to derive function-space empirical Bayes (FS-EB)—a method that combines parameter- and function-spaces regularization. We demonstrated that FS-EB exhibits desirable empirical properties, such as significantly improved predictive uncertainty quantification both in-distribution and under semantic shift. FS-EB is scalable, can be applied to any neural network architecture, can be used with pretrained models, and allows effectively incorporating prior information in a probabilistically principled manner.

Acknowledgments

We thank anonymous reviewers for useful feedback. This work is supported by NSF CAREER IIS-2145492, NSF I-DISRE 193471, NIH R01DA048764-01A1, NSF IIS-1910266, NSF 1922658 NRT-HDR, Meta Core Data Science, Google AI Research, BigHat Biosciences, Capital One, and an Amazon Research Award.

References

Asia Pacific Tele-Ophthalmology Society. Aptos 2019 blindness detection, 2019. URL https://www.kaggle.com/competitions/aptos2019-blindness-detection/overview.

Band, N., Rudner, T. G. J., Feng, Q., Filos, A., Nado, Z., Dusenberry, M. W., Jerfel, G., Tran, D., and Gal, Y. Benchmarking Bayesian Deep Learning on Diabetic Retinopathy Detection Tasks. In *Advances in Neural Information Processing Systems 34*, 2021.

Benjamin, A. S., Rolnick, D., and Kording, K. P. Measuring and regularizing networks in function space. *ArXiv*, abs/1805.08289, 2018.

Bietti, A. and Mairal, J. Group invariance, stability to deformations, and complexity of deep convolutional representations, 2018.

Bietti, A., Mialon, G., Chen, D., and Mairal, J. A kernel perspective for regularizing deep neural networks. In *International Conference on Machine Learning*, pp. 664–674. PMLR, 2019.

Bishop, C. M. Pattern recognition and machine learning (information science and statistics). 2006.

Breiman, L. Random forests. *Machine Learning*, 45:5–32, 2001.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T. J., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.

Burt, D. R., Ober, S., Garriga-Alonso, A., and van der Wilk, M. Understanding variational inference in function-space. *ArXiv*, abs/2011.09421, 2020.

Chen, Z., Shi, X., Rudner, T. G. J., Feng, Q., Zhang, W., and Zhang, T. A neural tangent kernel perspective on function-space regularization in neural networks. In *OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop)*, 2022.

Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., and Ha, D. Deep learning for classical japanese literature. *ArXiv*, abs/1812.01718, 2018.

Denker, J. S. and LeCun, Y. Transforming neural-net output levels to probability distributions. In *NIPS*, 1990.

El-Yaniv, R. and Wiener, Y. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(53):1605–1641, 2010.

Fang, A., Kornblith, S., and Schmidt, L. Does progress on imagenet transfer to real-world datasets? *ArXiv*, abs/2301.04644, 2023.

Ha, Q., Liu, B., and Liu, F. Identifying melanoma images using efficientnet ensemble: Winning solution to the SIIM-ISIC melanoma classification challenge. *CoRR*, abs/2010.05351, 2020.

Hanke, J. 1st place solution, 2021. URL https://www.kaggle.com/competitions/cassava-leaf-disease-classification/discussion/221957.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778. IEEE Computer Society, 2016.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

Joo, T. and Chung, U. Revisiting explicit regularization in neural networks for well-calibrated predictive uncertainty, 2020.

Krizhevsky, A. Convolutional deep belief networks on cifar-10. 2010.

Krogh, A. and Hertz, J. A. A simple weight decay can improve generalization. In *NIPS*, 1991.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6402–6413, 2017.

Ma, C. and Hernández-Lobato, J. M. Functional variational inference based on stochastic process generators. In *NeurIPS*, 2021.

Ma, C., Li, Y., and Hernández-Lobato, J. M. Variational implicit processes. In *International Conference on Machine Learning*, 2018.

Murphy, K. P. *Machine learning : a probabilistic perspective*. MIT Press, Cambridge, Mass. [u.a.], 2013. ISBN 9780262018029 0262018020.

Mwebaze, E., Gebru, T., Frome, A., Nsumba, S., and Tusubira, J. icassava 2019fine-grained visual categorization challenge, 2019.

Nado, Z., Band, N., Collier, M., Djolonga, J., Dusenberry, M. W., Farquhar, S., Filos, A., Havasi, M., Jenatton, R., Jerfel, G., Liu, J., Mariet, Z., Nixon, J., Padhy, S., Ren, J., Rudner, T. G. J., Wen, Y., Wenzel, F., Murphy, K., Sculley, D., Lakshminarayanan, B., Snoek, J., Gal, Y., and Tran, D. Uncertainty Baselines: Benchmarks for Uncertainty & Robustness in Deep Learning. 2021.

Naeini, M. P., Cooper, G. F., and Hauskrecht, M. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, 2015: 2901–2907, 2015.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Reading digits in natural images with unsupervised feature learning. 2011.

Ober, S. and Aitchison, L. Global inducing point variational posteriors for bayesian neural networks and deep gaussian processes. In *International Conference on Machine Learning*, 2020.

Pan, I. [2nd place] solution overview, 2020. URL https://www.kaggle.com/competitions/siim-isic-melanoma-classification/discussion/175324.

Rabanser, S., Thudi, A., Hamidieh, K., Dziedzic, A., and Papernot, N. Selective classification via neural network training dynamics, 2022.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do cifar-10 classifiers generalize to cifar-10? 2018.

Rudner, T. G. J., Chen, Z., Teh, Y. W., and Gal, Y. Tractable function-space variational inference in Bayesian neural networks. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022a.

Rudner, T. G. J., Smith, F. B., Feng, Q., Teh, Y. W., and Gal, Y. Continual Learning via Sequential Function-Space Variational Inference. In *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2022b.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. S., Berg, A. C., and Fei-Fei, L. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2014.

SIIM and ISIC. Siim-isic melanoma classification, 2020. URL https://www.kaggle.com/competitions/siim-isic-melanoma-classification/overview.

Sun, S., Zhang, G., Shi, J., and Grosse, R. B. Functional variational bayesian neural networks. *ArXiv*, abs/1903.05779, 2019a.

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

Titsias, M. K., Schwarz, J., de G. Matthews, A. G., Pascanu, R., and Teh, Y. W. Functional regularisation for continual learning using gaussian processes. *ArXiv*, abs/1901.11356, 2019.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023.

Tran, D., Liu, J., Dusenberry, M. W., Phan, D., Collier, M., Ren, J., Han, K., Wang, Z., Mariet, Z., Hu, H., Band, N., Rudner, T. G. J., Singhal, K., Nado, Z., van Amersfoort andAndreas Kirsch, J., Jenatton, R., Thain, N., Yuan, H., Buchanan, K., Murphy, K., Sculley, D., Gal, Y., Ghahramani, Z., Snoek, J., and Lakshminarayanan, B. Plex: Towards Reliability Using Pretrained Large Model Extensions. In *ICML 2022 Workshop on Pre-training: Perspectives, Pitfalls, and Paths Forward*, 2022.

Wang, Z., Ren, T., Zhu, J., and Zhang, B. Function space particle optimization for bayesian neural networks. *ArXiv*, abs/1902.09754, 2019.

Wilson, A. G. and Izmailov, P. Bayesian deep learning and a probabilistic perspective of generalization. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Wolpert, D. H. Bayesian backpropagation over i-o functions rather than weights. In Cowan, J., Tesauro, G., and Alspector, J. (eds.), *Advances in Neural Information Processing Systems*, volume 6. Morgan-Kaufmann, 1993.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. 2017.

Xu, G. 1st place solution summary, 2019. URL https://www.kaggle.com/competitions/aptos2019-blindness-detection/discussion/108065.

Appendix

A. Additional Details and Experiments

A.1. Hyperparameters

In Table 5, we provide the key hyperparameters used with FS-EB. We operate over the search space using randomized grid search. In addition to the learning rate η, cosine scheduler α, and weight decay used by standard PS-MAP, we use two more hyperparameters—the prior variance τ_f^{-1} and the number of Monte Carlo samples J.

Table 5: Hyperparameter Ranges

HYPERPARAMETER	RANGE
LEARNING RATE η	$[10^{-10}, 10^{-1}]$
SCHEDULER α	$[0, 1]$
WEIGHT DECAY τ_θ^{-1}	$[10^{-10}, 1]$
PRIOR VARIANCE τ_f^{-1}	$[10^{-7}, 5 \times 10^4]$
MONTE CARLO SAMPLES J	$\{1, 2, 5, 10\}$

A.2. Deep Ensembles

Lakshminarayanan et al. (2017) propose a simple alternative to Bayesian neural networks by computing the Bayesian model average using a set of independently trained neural networks, i.e. the softmax outputs from each independent network are averaged to provide the final predictive distribution for classification. This method is called Deep Ensembles. Across literature, Deep Ensembles have been observed to provide improved generalization and better calibration. Subsequently, in Table 6, we quantify the benefit of Deep Ensembles for FS-EB. Surprisingly, we find that Deep Ensembles benefit PS-MAP more than they do FS-EB. A key property of ensemble components that lead to better generalization is the induced diversity (Breiman, 2001). We speculate that FS-EB may enforce a bias that makes the components of an ensemble less diverse, since it has a more informative prior than standard weight decay.

Table 6: We report the accuracy (ACC.), negative log-likelihood (NLL), expected calibration error (ECE), area under selective prediction accuracy curve (SEL. PRED.), and area under OOD prediction accuracy curve (OOD) for FashionMNIST (Xiao et al., 2017) and CIFAR-10 (Krizhevsky, 2010) with FS-EB deep ensembles (Lakshminarayanan et al., 2017).

	FASHIONMNIST					CIFAR-10				
METHOD	ACC. ↑	SEL. PRED. ↑	NLL ↓	ECE ↓	OOD ↑	ACC. ↑	SEL. PRED. ↑	NLL ↓	ECE ↓	OOD ↑
PS-MAP-ENSEMBLE	94.5%	99.3%	**0.18**	**1.6%**	94.9%	**96.0%**	**99.6%**	**0.13**	**0.7%**	95.7%
FS-EB-ENSEMBLE	**94.7%**	**98.9%**	0.21	3.7%	**99.9%**	95.8%	99.5%	0.17	3.0%	**99.1%**

A.3. Performance with CIFAR-10.1

Recht et al. (2018) introduce an extended set of test samples similar in distribution to CIFAR-10 meant as a safeguard against overfitting of methods to benchmark classification task of CIFAR-10. In Table 7, we report the performance metrics for CIFAR-10 trained models evaluated on the CIFAR-10.1 test set.

Table 7: We report the accuracy (ACC.), negative log-likelihood (NLL), expected calibration error (ECE), area under selective prediction accuracy curve (SEL. PRED.), and area under OOD prediction accuracy curve (OOD) for CIFAR-10.1 (Recht et al., 2018) using models trained on CIFAR-10. Means and standard errors are computed over five seeds.

METHOD	ACC. ↑	SEL. PRED. ↑	NLL ↓	ECE ↓
PS-MAP	$\mathbf{88.0\%} \pm 0.1$	$97.5\% \pm 0.1$	0.49 ± 0.00	$7.6\% \pm 0.1$
FS-EB	$86.8\% \pm 0.4$	$97.2\% \pm 0.2$	0.49 ± 0.01	$\mathbf{4.0\%} \pm 0.2$

A.4. Model Robustness with CIFAR-10 Corrupted

Hendrycks & Dietterich (2019) propose the CIFAR-10 Corrupted dataset as a test for model robustness, which consists of 19 commonly observed corruptions of images including blur, noise, and pixelation. All corruptions are created with CIFAR-10 test images at five different levels.

In continuation of the discussion around Figure 3, we summarize the accuracy and selective accuracy across all the corruptions in Figures 4 and 5.

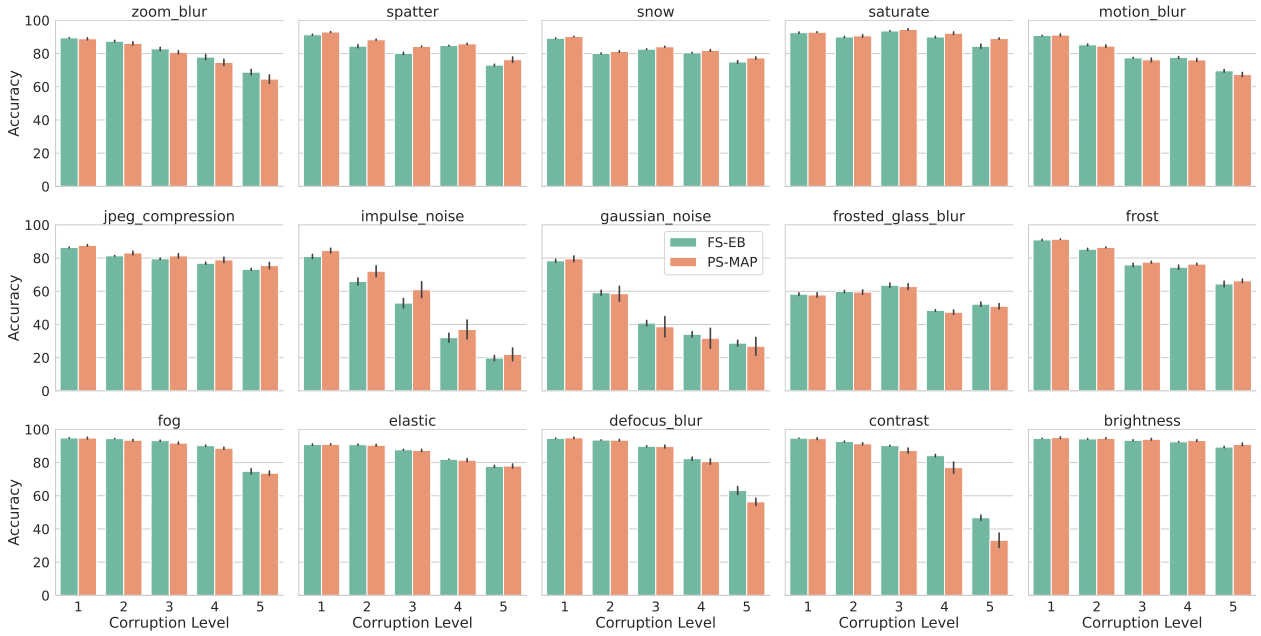


Figure 4: Accuracy on CIFAR-10 Corrupted

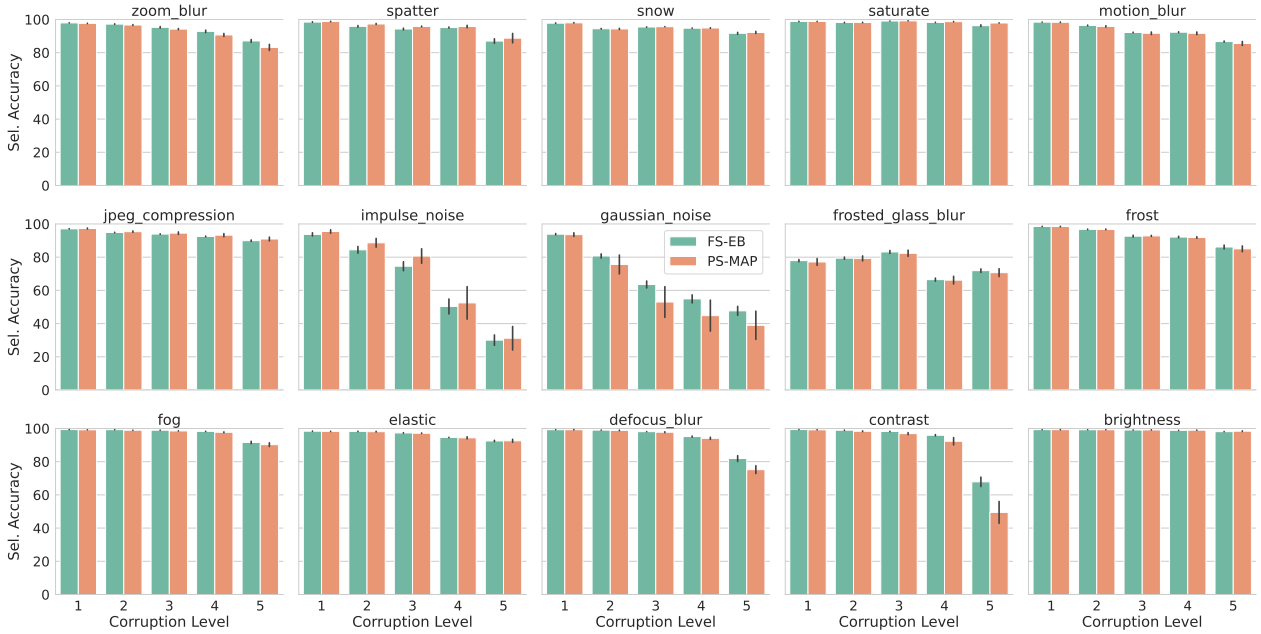


Figure 5: Selective Accuracy on CIFAR-10 Corrupted

A.5. Effect of Training Data Size

In Tables 8 and 9, we quantify the performance of FS-EB in the low-data regime. For various fractions (10%, 25%, 50%, 75%) of the full training dataset, we train both PS-MAP and FS-EB. Across all metrics, we find that FS-EB overall tends to outperform PS-MAP significantly.

Table 8: We assess the performance of FS-EB in the low training data regime for FashionMNIST. Overall, we find that FS-EB tends to generalize significantly better under small data, similar to our findings for FashionMNIST in Table 9. Means and standard errors are computed over five seeds.

FRACTION	METHOD	ACC. ↑	SEL. PRED. ↑	NLL ↓	ECE ↓	OOD AUROC ↑
10%	FS-EB	**89.0**%±0.1	**97.2**%±0.1	**0.47**±0.01	**6.7**%±0.1	**98.1**%±0.4
	PS-MAP	88.1%±0.2	97.0%±0.1	0.49±0.00	7.4%±0.1	88.1%±2.1
25%	FS-EB	**91.5**%±0.1	98.0%±0.1	**0.35**±0.01	**5.2**%±0.1	**98.6**%±0.2
	PS-MAP	91.1%±0.1	**98.3**%±0.0	0.36±0.00	5.4%±0.1	88.6%±1.3
50%	FS-EB	**92.9**%±0.0	98.2%±0.1	0.31±0.00	**4.6**%±0.1	**99.5**%±0.1
	PS-MAP	92.5%±0.1	**98.7**%±0.0	**0.30**±0.01	4.5%±0.1	93.0%±0.2
75%	FS-EB	**93.6**%±0.1	98.3%±0.0	0.29±0.00	4.4%±0.1	**99.8**%±0.0
	PS-MAP	93.2%±0.1	**98.9**%±0.0	**0.28**±0.00	**4.2**%±0.1	93.1%±0.7
100%	FS-EB	**94.1**%±0.1	98.8%±0.0	**0.19**±0.00	**1.8**%±0.1	**99.9**%±0.0
	PS-MAP	93.8%±0.0	**98.9**%±0.0	0.26±0.00	3.6%±0.0	94.9%±0.4
	PS-MAP	91.1%±0.1	**98.3**%±0.0	0.36±0.00	5.4%±0.1	88.6%±1.3

Table 9: We assess the performance of FS-EB in the low training data regime for CIFAR-10. Overall, we find that FS-EB tends to generalize significantly better under small data, similar to our findings for FashionMNIST in Table 8. Means and standard errors are computed over five seeds.

FRACTION	METHOD	ACC. ↑	SEL. PRED. ↑	NLL ↓	ECE ↓	OOD AUROC ↑
10%	FS-EB	**78.3**%±0.1	**93.2**%±0.0	**0.83**±0.00	**11.1**%±0.3	**95.9**%±0.3
	PS-MAP	72.7%±0.1	89.9%±0.1	1.36±0.00	19.7%±0.0	66.2%±1.0
25%	FS-EB	**87.6**%±0.0	**97.2**%±0.0	**0.47**±0.00	**6.0**%±0.1	**99.6**%±0.0
	PS-MAP	87.1%±0.4	97.1%±0.1	0.54±0.01	7.9%±0.2	74.8%±2.5
50%	FS-EB	92.0%±0.1	98.7%±0.0	**0.30**±0.00	**2.6**%±0.1	**99.9**%±0.0
	PS-MAP	**92.5**%±0.0	98.7%±0.0	0.32±0.01	4.7%±0.1	85.9%±1.3
75%	FS-EB	93.9%±0.1	99.1%±0.0	0.23±0.0	**1.8**%±0.0	**99.9**%±0.0
	PS-MAP	**94.4**%±0.0	99.1%±0.0	0.23±0.00	3.4%±0.0	91.6%±0.8
100%	FS-EB	**95.1**%±0.1	**99.4**%±0.0	**0.20**±0.00	**2.1**%±0.1	**99.4**%±0.0
	PS-MAP	94.9%±0.1	99.3%±0.0	0.21±0.01	3.0%±0.0	93.0%±0.2

A.6. Effect of Context Set Batch Size

During each gradient step of FS-EB training, we use a subset of points from the context distribution, sampled uniformly at random as described in Section 3. The number of samples is what we call the context set batch size. In Table 10, we vary this batch size and find that most metrics are not very sensitive to this hyperparameter choice.

Table 10: We vary the size of the context set batch size ad assess the effect on predictive performance.

	FASHIONMNIST					CIFAR-10				
BATCH SIZE	ACC. ↑	SEL. PRED. ↑	NLL ↓	ECE ↓	OOD ↑	ACC. ↑	SEL. PRED. ↑	NLL ↓	ECE ↓	OOD ↑
32	94.1% ± 0.0	**98.4**% ± 0.1	**0.27** ± 0.00	**4.1**% ± 0.0	98.9% ± 0.1	95.0% ± 0.1	99.3% ± 0.0	0.19 ± 0.00	1.5% ± 0.1	99.9% ± 0.0
64	94.1% ± 0.0	98.3% ± 0.0	**0.27** ± 0.00	**4.1**% ± 0.0	99.5% ± 0.0	94.9% ± 0.1	99.3% ± 0.0	0.19 ± 0.0	**1.4**% ± 0.0	99.9% ± 0.0
128	94.1% ± 0.0	98.3% ± 0.0	0.28 ± 0.00	4.2% ± 0.0	**99.9**% ± 0.0	**95.1**% ± 0.1	**99.4**% ± 0.0	0.20 ± 0.00	2.1% ± 0.1	99.4% ± 0.0

A.7. Effect of Training Context Distribution

We study the effect of different context set distributions. In our main experiments, we use KMNIST (Clanuwat et al., 2018) as the context distribution for FashionMNIST and CIFAR-100 as the context distribution for CIFAR-10. In Table 11, we evaluate the performance of FS-EB with the context set being (i) the training inputs and (ii) corrupted training inputs.

Table 11: We vary the context set (CTX. SET) distribution to be (i) the training set, and (ii) the training set with data augmentations and quantify the performance of FS-EB. $\mathbf{X}_C$ = KMNIST for FashionMNIST and $\mathbf{X}_C$ = CIFAR-100 for CIFAR-10. Changing the context set distribution does have a significant impact on generalization performance in terms of accuracy and can also lead to significant improvement in out-of-distribution detection.

	FASHIONMNIST					CIFAR-10				
CTX. SET	ACC. ↑	SEL. PRED. ↑	NLL ↓	ECE ↓	OOD ↑	ACC. ↑	SEL. PRED. ↑	NLL ↓	ECE ↓	OOD ↑
TRAIN	93.9%±0.0	98.3%±0.1	0.28±0.00	4.2%±0.0	97.6%±0.5	94.9%±0.1	99.3%±0.0	**0.19**±0.00	1.7%±0.1	92.1%±0.6
TRAIN CORR.	**94.1**%±0.0	98.4%±0.0	0.27±0.00	4.1%±0.0	97.7%±0.5	94.7%±0.1	99.2%±0.0	0.20±0.00	**1.4**%±0.0	99.9%±0.0
$\mathbf{X}_C$	**94.1**%±0.1	**98.8**%±0.0	**0.19**±0.00	**1.8**%±0.1	**99.9**%±0.0	**95.1**%±0.1	**99.4**%±0.0	0.20±0.00	2.1%±0.1	**99.4**%±0.1

A.8. Transfer Learning on Real-World Datasets

In addition to standard benchmark datasets, we also consider three additional real-world datasets - APTOS Blindness Detection (Asia Pacific Tele-Ophthalmology Society, 2019; Xu, 2019), Melanoma Classification (SIIM & ISIC, 2020; Ha et al., 2020; Pan, 2020), and Cassava Leaf Disease Classification (Mwebaze et al., 2019; Hanke, 2021)

Table 12: Performance on Real-World Datasets, transfer learning from an ImageNet-pretrained ResNet-50 (He et al., 2016).

DATASET	METHOD	ACC. ↑	SEL. PRED. ↑	NLL ↓	ECE ↓
APTOS	FS-EB	83.2%	94.2%	**0.78**	**11.3**%
	PS-MAP	**83.7**%	**93.7**%	0.83	12.8%
MELANOMA	FS-EB	**98.6**%	**99.8**%	**0.05**	**1.6**%
	PS-MAP	98.2%	99.7%	0.08	1.8%
CASSAVA	FS-EB	86.5%	**96.5**%	**0.64**	**9.0**%
	PS-MAP	86.5%	95.6%	0.80	10.9%

Using an ImageNet-pretrained (Russakovsky et al., 2014) ResNet-50 (He et al., 2016), similar in spirit to Fang et al. (2023), we conduct a transfer learning experiment. In Table 12, we provide the performance of FS-EB on these datasets and find that FS-EB can often provide improvements in the data fit in terms of the data likelihood and much better calibration in terms of ECE (Naeini et al., 2015).

A.9. Runtimes

For reference, we provide approximate runtimes of FS-EB and PS-MAP in Table 13.

Table 13: Approximate runtime for a single gradient step and one full epoch of training for FashionMNIST and CIFAR-10.

Dataset	Method	Gradient Step (ms) ↓	Epoch (s) ↑
FashionMNIST	PS-MAP	40	18
	FS-EB	129	60
	FS-VI	319	144
CIFAR-10	PS-MAP	55	21
	FS-EB	137	61
	FS-VI	389	189