# FSP-LAPLACE: Function-Space Priors for the Laplace Approximation in Bayesian Deep Learning

**Tristan Cinquin**\*
Tübingen AI Center, University of Tübingen
`tristan.cinquin@uni-tuebingen.de`

**Marvin Pförtner**\*
Tübingen AI Center, University of Tübingen
`marvin.pfoertner@uni-tuebingen.de`

**Vincent Fortuin**
Helmholtz AI, TU Munich
`vincent.fortuin@tum.de`

**Philipp Hennig**
Tübingen AI Center, University of Tübingen
`philipp.hennig@uni-tuebingen.de`

**Robert Bamler**
Tübingen AI Center, University of Tübingen
`robert.bamler@uni-tuebingen.de`

## Abstract

Laplace approximations are popular techniques for endowing deep networks with epistemic uncertainty estimates as they can be applied without altering the predictions of the trained network, and they scale to large models and datasets. While the choice of prior strongly affects the resulting posterior distribution, computational tractability and lack of interpretability of the weight space typically limit the Laplace approximation to isotropic Gaussian priors, which are known to cause pathological behavior as depth increases. As a remedy, we directly place a prior on function space. More precisely, since Lebesgue densities do not exist on infinite-dimensional function spaces, we recast training as finding the so-called weak mode of the posterior measure under a Gaussian process (GP) prior restricted to the space of functions representable by the neural network. Through the GP prior, one can express structured and interpretable inductive biases, such as regularity or periodicity, directly in function space, while still exploiting the implicit inductive biases that allow deep networks to generalize. After model linearization, the training objective induces a negative log-posterior density to which we apply a Laplace approximation, leveraging highly scalable methods from matrix-free linear algebra. Our method provides improved results where prior knowledge is abundant (as is the case in many scientific inference tasks). At the same time, it stays competitive for black-box supervised learning problems, where neural networks typically excel.

## 1 Introduction

Neural networks (NNs) have become the workhorse for many machine learning tasks, but they do not quantify the uncertainty arising from data scarcity—the *epistemic* uncertainty. NNs therefore cannot estimate their confidence in their predictions, which is needed for safety-critical applications, decision making, and scientific modeling [1–3]. As a solution, Bayesian neural networks (BNNs) cast training as approximating the Bayesian posterior distribution over the weights (i.e., the distribution of the model parameters given the training data), thus naturally capturing epistemic uncertainty. Various methods exist for approximating the (intractable) posterior, either by a set of samples (e.g., MCMC) or

---

\*Equal contribution

(a) FSP-LAPLACE posterior predictive of a BNN under different choices of GP prior



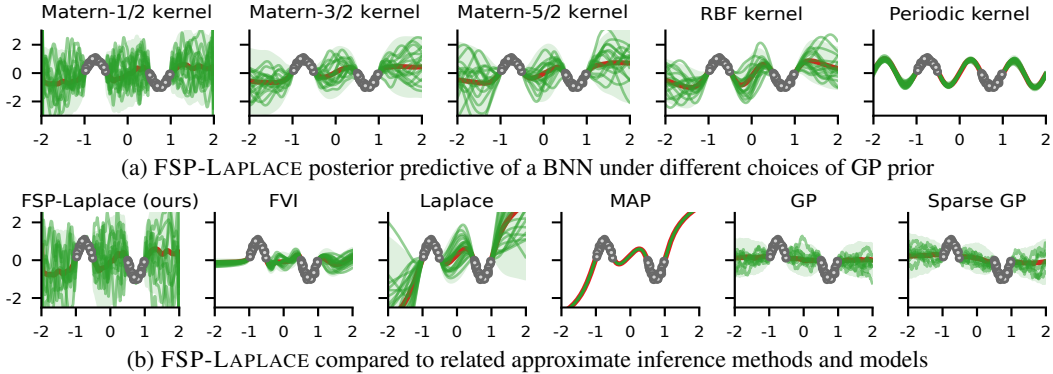(b) FSP-LAPLACE compared to related approximate inference methods and models

Figure 1: FSP-LAPLACE allows for efficient approximate Bayesian neural network (BNN) inference under interpretable function space priors. Using our method, it is possible to encode functional properties like smoothness, lengthscale, or periodicity through a Gaussian process (GP) prior. The gray data points in the plots are noisy observations of a periodic function.

by a parametric distribution (e.g., variational inference or Laplace approximations). While sampling methods can be asymptotically exact for large numbers of samples, they are typically expensive for large models, whereas variational inference (VI) and the Laplace approximation are more scalable. The Laplace approximation is particularly appealing as it does not alter maximum-a-posteriori (MAP) predictions. Compared to VI, its differential nature allows the Laplace approximation to scale to large datasets and models, and it has been shown to provide well-calibrated uncertainty estimates [4, 5].

**The need for function-space priors in BNNs.** While the choice of prior strongly influences the uncertainty estimates obtained from the posterior [6, 7], there are hardly any methods for elicitation of informative priors in the literature on BNNs, with the notable exceptions of Sun et al. [8] and Tran et al. [9]. This is not just a conceptual issue. For instance, the default isotropic Gaussian prior, commonly thought of as uninformative, actually carries incorrect beliefs about the posterior (uni-modality, independent weights, etc.) [6, 10] and has known pathologies in deep NNs [9]. As network weights are not interpretable, formulating a good prior on them is virtually impossible. Current methods work around this issue by model selection, either through expensive cross-validation, or type-II maximum likelihood estimation with a Laplace approximation under an isotropic Gaussian prior, which can only be computed exactly for small networks, and has known issues with normalization layers [11, 12]. As a solution, Sun et al. [8] proposed a VI method to specify priors directly on the function implemented by the BNN, with promising results. Function-space priors incorporate interpretable knowledge about the variance, regularity, periodicity, or length scale, building on the extensive Gaussian Process (GP) literature [13]. But it turns out that the method by Sun et al. [8] is difficult to use in practice: the Kullback-Leibler (KL) divergence regularizing VI is infinite for most function-space priors of interest [14], and estimating finite variants of it is challenging [15].

This paper addresses the lack of a procedure to specify informative priors in the Laplace approximation by proposing a method to use interpretable GP priors. Motivated by MAP estimation theory, we first derive an objective function that regularizes the neural network in function space using a GP prior, and whose minimizer corresponds to the MAP estimator of a GP on the space of functions represented by the network. We then apply the Laplace approximation to the log-posterior density induced by the objective, allowing us to effectively incorporate beliefs from a GP prior into a deep network (see Figure 1). Efficient matrix-free methods from numerical linear algebra allow scaling our method to large models and datasets. We show the effectiveness of our method by showing improved results in cases where prior knowledge is available, and competitive performance for black-box regression and classification tasks, where neural networks typically excel. We make the following contributions:

1. We propose a novel objective function for deep neural networks that allows incorporating interpretable prior knowledge through a GP prior in function space.

2. We develop an efficient and scalable Laplace approximation that endows the neural network with epistemic uncertainty reflecting the beliefs specified by the GP prior in function space.

3. A range of experiments shows that our method improves performance on tasks with prior knowledge in the form of a kernel, while showing competitive performance for black-box regression, classification, out-of-distribution detection, and Bayesian optimization tasks.

2

## 2 Preliminaries: Laplace approximation in weight space

For a given dataset $\mathbb{D} = (\boldsymbol{X}, \boldsymbol{Y})$ of inputs $\boldsymbol{X} = (\boldsymbol{x}^{(i)})_{i=1}^n \in \mathbb{X}^n$ and targets $\boldsymbol{Y} = (\boldsymbol{y}^{(i)})_{i=1}^n \in \mathbb{Y}^n$, we consider a model of the data that is parameterized by a neural network $\boldsymbol{f} \colon \mathbb{X} \times \mathbb{W} \to \mathbb{R}^{d'}$ with weights $\boldsymbol{w} \in \mathbb{W} \subset \mathbb{R}^p$, a likelihood $p(\boldsymbol{Y} \mid \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{w})) \coloneqq \prod_{i=1}^n p(\boldsymbol{y}^{(i)} \mid \boldsymbol{f}(\boldsymbol{x}^{(i)}, \boldsymbol{w}))$, and a prior $p(\boldsymbol{w})$. We seek the Bayesian posterior $p(\boldsymbol{w} \mid \mathbb{D}) \propto p(\boldsymbol{Y} \mid \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{w})) \, p(\boldsymbol{w})$. As it is intractable in BNNs, approximate inference methods have been developed. Among these, the Laplace approximation [16] to the posterior is given by a Gaussian distribution $q(\mathbf{w} = \boldsymbol{w}) = \mathcal{N}\left(\boldsymbol{w}; \boldsymbol{w}^\star, \boldsymbol{\Lambda}^{-1}\right)$, whose parameters are found by MAP estimation $\boldsymbol{w}^\star \in \arg\min_{\boldsymbol{w} \in \mathbb{W}} R_{\mathrm{WS}}(\boldsymbol{w})$ of the weights, where

$$R_{\mathrm{WS}}(\boldsymbol{w}) \coloneqq -\log p(\boldsymbol{w} \mid \mathbb{D}) = -\log p(\boldsymbol{w}) - \sum_{i=1}^n \log p(\boldsymbol{y}^{(i)} \mid \boldsymbol{f}(\boldsymbol{x}^{(i)}, \boldsymbol{w})) + \text{const.} \quad (2.1)$$

and computing the Hessian of the negative log-posterior $\boldsymbol{\Lambda} \coloneqq -\left.\mathbf{H}_{\boldsymbol{w}} \log p(\boldsymbol{w} \mid \mathbb{D})\right|_{\boldsymbol{w}=\boldsymbol{w}^\star} \in \mathbb{R}^{p \times p}$.

**The linearized Laplace approximation.** Computing $\boldsymbol{\Lambda}$ involves the Hessian of the neural network w.r.t. its weights which is generally expensive. To make the Laplace approximation scalable, it is common to linearize the network around $\boldsymbol{w}^\star$ before computing $\boldsymbol{\Lambda}$ [17],

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{w}) \approx \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{w}^\star) + \boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{x})(\boldsymbol{w} - \boldsymbol{w}^\star) =: \boldsymbol{f}^{\mathrm{lin}}(\boldsymbol{x}, \boldsymbol{w}) \quad (2.2)$$

with Jacobian $\boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{x}) = \left.\mathbf{D}_{\boldsymbol{w}} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{w})\right|_{\boldsymbol{w}=\boldsymbol{w}^\star}$. Thus, the approximate posterior precision matrix $\boldsymbol{\Lambda}$ is

$$\boldsymbol{\Lambda} \approx -\left.\mathbf{H}_{\boldsymbol{w}} \log p(\boldsymbol{w})\right|_{\boldsymbol{w}=\boldsymbol{w}^\star} - \sum_{i=1}^n \boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{x}^{(i)})^\top \boldsymbol{H}_{\boldsymbol{w}^\star}^{(i)} \boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{x}^{(i)}), \quad (2.3)$$

where $\boldsymbol{H}_{\boldsymbol{w}^\star}^{(i)} = \left.\mathbf{H}_{\boldsymbol{f}} \log p(\boldsymbol{y}^{(i)} \mid \boldsymbol{f})\right|_{\boldsymbol{f}=\boldsymbol{f}(\boldsymbol{x}^{(i)}, \boldsymbol{w}^\star)}$. Thus, under the linearized network, the Hessian of the negative log-likelihood (NLL) coincides with the *generalized Gauss-Newton matrix* (GGN) of the NLL. Crucially, the GGN is positive-(semi)definite even if $-\left.\mathbf{H}_{\boldsymbol{w}} \log p(\boldsymbol{Y} \mid \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{w}))\right|_{\boldsymbol{w}=\boldsymbol{w}^\star}$ is not.

## 3 FSP-LAPLACE: Laplace approximation under function-space priors

A conventional Laplace approximation in neural networks requires a prior in weight space, with the issues discussed in Section 1. We now present FSP-LAPLACE, a method for computing Laplace approximations under interpretable GP priors in function space. Section 3.1 introduces an objective function that is a log-density under local linearization. Section 3.2 proposes a scalable algorithm for the linearized Laplace approximation with a function-space prior using matrix-free linear algebra.

### 3.1 Laplace approximations in function space

We motivate our Laplace approximation in function space through the lens of MAP estimation in an (infinite-dimensional) function space under a GP prior. As Lebesgue densities do not exist in (infinite-dimensional) function spaces, we cannot minimize Equation (2.1) to find the MAP estimate. We address this issue using a generalized notion of MAP estimation, resulting in a minimizer of a regularized objective on the reproducing kernel Hilbert space $\mathbb{H}_{\boldsymbol{\Sigma}}$. We then constrain the objective to the set of functions representable by the neural network and minimize it using tools from deep learning. Finally, local linearization of the neural network turns this objective into a valid negative log-density, from which we obtain the posterior covariance by computing its Hessian.

**MAP estimation in neural networks under Gaussian process priors.** The first step of the Laplace approximation is to find a MAP estimate of the neural network weights $\boldsymbol{w}$, i.e., the minimizer of the negative log-density function in Equation (2.1) w.r.t. the Lebesgue measure as a "neutral" reference measure. In our method, we regularize the neural network in function space using a $d'$-output GP prior $\mathbf{f} \sim \mathcal{GP}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ with index set $\mathbb{X}$. However, a (nonparametric) GP takes its values in an infinite-dimensional (Banach) space $\mathbb{B}$ of functions, where no such reference Lebesgue measure exists [18]. Rather, the GP induces a prior *measure* $\mathrm{P}_{\mathbb{B}}$ on $\mathbb{B}$ [19, Section B.2]. We are thus interested in the "mode" of the posterior measure $\mathrm{P}_{\mathbb{B}}^{\boldsymbol{Y}}$ under $\mathrm{P}_{\mathbb{B}}$ defined by the Radon-Nikodym

derivative $\mathrm{P}_{\mathbb{B}}^{\boldsymbol{Y}}(\mathrm{d}\boldsymbol{f}) \propto \exp\left(-\Phi^{\boldsymbol{Y}}(\mathrm{d}\boldsymbol{f})\right) \mathrm{P}_{\mathbb{B}}(\mathrm{d}\boldsymbol{f})$ where $\Phi^{\boldsymbol{Y}}$ is the *potential*, in essence the negative log-likelihood functional of the model. In our case, $\Phi^{\boldsymbol{Y}}(\boldsymbol{f}) = -\sum_{i=1}^{n} \log p(\boldsymbol{y}^{(i)} \mid \boldsymbol{f}(\boldsymbol{x}^{(i)}))$. Similar to Bayes' rule in finite dimensions, this Radon-Nikodym derivative relates the prior measure to the posterior measure by reweighting. Since there is no Lebesgue measure, the (standard) mode is undefined, and we therefore follow Lambley [20], using so-called *weak modes*.

**Definition 1** (Weak Mode [see e.g., 20, Definition 2.1]). Let $\mathbb{B}$ be a separable Banach space and let P be a probability measure on $(\mathbb{B}, \mathcal{B}(\mathbb{B}))$. A *weak mode* of P is any point $\boldsymbol{f}^\star \in \operatorname{supp} \mathrm{P}$ such that

$$\limsup_{r \downarrow 0} \frac{\mathrm{P}(B_r(\boldsymbol{f}))}{\mathrm{P}(B_r(\boldsymbol{f}^\star))} \leq 1 \qquad \text{for all } \boldsymbol{f} \in \mathbb{B}. \tag{3.1}$$

Above, $\mathcal{B}(\mathbb{B})$ denotes the Borel $\sigma$-algebra on $\mathbb{B}$ and $B_r(\boldsymbol{f}) \subset \mathbb{B}$ is an open ball with radius $r$ centered at a point $\boldsymbol{f} \in \mathbb{B}$. The intuition for the weak mode is the same as for the finite-dimensional mode (indeed they coincide when a Lebesgue measure exists), but the weak mode generalizes this notion to infinite-dimensional (separable) Banach spaces. Under certain technical assumptions on the potential $\Phi^{\boldsymbol{Y}}$ (see Assumption A.2 in the appendix), Lambley [20] shows that any solution to

$$\operatorname*{arg\,min}_{\boldsymbol{f} \in \mathbb{H}_{\boldsymbol{\Sigma}}} \underbrace{\Phi^{\boldsymbol{Y}}(\boldsymbol{f}) + \frac{1}{2}\|\boldsymbol{f} - \boldsymbol{\mu}\|_{\mathbb{H}_{\boldsymbol{\Sigma}}}^2}_{=:R_{\mathrm{FSP}}(\boldsymbol{f})} \tag{3.2}$$

is a weak mode of the posterior probability measure $\mathrm{P}_{\mathbb{B}}^{\boldsymbol{Y}}$.

Equation (3.2) casts the weak mode of the posterior measure $\mathrm{P}_{\mathbb{B}}^{\boldsymbol{Y}}$ as the solution of an optimization problem in the RKHS $\mathbb{H}_{\boldsymbol{\Sigma}}$. We can now relate it to an optimization problem in weight space $\mathbb{W}$, and apply tools from deep learning. Informally speaking, we assume that the intersection of the set of functions represented by the neural network $\mathbb{F} := \{\boldsymbol{f}(\cdot, \boldsymbol{w}): \boldsymbol{w} \in \mathbb{W}\} \subset \mathbb{B} \subset (\mathbb{R}^{d'})^{\mathbb{X}}$ and $\mathbb{H}_{\boldsymbol{\Sigma}}$ is non-empty, and constrain the optimization problem in Equation (3.2) such that $\boldsymbol{f}$ belongs to both sets

$$\operatorname*{arg\,min}_{\boldsymbol{f} \in \mathbb{H}_{\boldsymbol{\Sigma}} \cap \mathbb{F}} R_{\mathrm{FSP}}(\boldsymbol{f}). \tag{3.3}$$

Unfortunately, the framework by Lambley [20] cannot give probabilistic meaning to optimization problems with hard constraints of the form $\boldsymbol{f} \in \mathbb{F}$. To address this, we adopt and elaborate on the informal strategy from Chen et al. [21, Remark 2.4]. Denote by $d_{\mathbb{B}}(\boldsymbol{f}, \mathbb{F}) := \inf_{\boldsymbol{f}_0 \in \mathbb{F}} \|\boldsymbol{f}_0 - \boldsymbol{f}\|_{\mathbb{B}}$ the distance of a function $\boldsymbol{f} \in \mathbb{B}$ to the set $\mathbb{F} \subset \mathbb{B}$. Under Assumption A.3, $d_{\mathbb{B}}(\boldsymbol{f}, \mathbb{F}) = 0$ if and only if $\boldsymbol{f} \in \mathbb{F}$ by Lemma A.2. Hence, we can relax the constraint $\boldsymbol{f} \in \mathbb{F}$ by adding $\frac{1}{2\lambda^2} d_{\mathbb{B}}^2(\boldsymbol{f}, \mathbb{F})$ with $\lambda > 0$ to the objective in Equation (3.2). Intuitively, the resulting optimization problem is the MAP problem for the measure $\mathrm{P}_{\mathbb{B}}^{\boldsymbol{Y},\lambda}$ obtained by conditioning $\mathrm{P}_{\mathbb{B}}^{\boldsymbol{Y}}$ on the observation that $d_{\mathbb{B}}(\mathbf{f}, \mathbb{F}) + \epsilon_\lambda = 0$, where $\epsilon_\lambda$ is independent centered Gaussian measurement noise with variance $\lambda^2$. Formally:

**Proposition 1.** *Let Assumptions A.1 to A.3 hold. For $\lambda > 0$, define $\Phi^{\boldsymbol{Y},\lambda}: \mathbb{B} \to \mathbb{R}, \boldsymbol{f} \mapsto \Phi^{\boldsymbol{Y}}(\boldsymbol{f}) + \frac{1}{2\lambda^2} d_{\mathbb{B}}^2(\boldsymbol{f}, \mathbb{F})$. Then the posterior measure $\mathrm{P}_{\mathbb{B}}^{\boldsymbol{Y},\lambda}(\mathrm{d}\boldsymbol{f}) \propto \exp\left(-\Phi^{\boldsymbol{Y},\lambda}(\mathrm{d}\boldsymbol{f})\right) \mathrm{P}_{\mathbb{B}}(\mathrm{d}\boldsymbol{f})$ has at least one weak mode $\boldsymbol{f}^\star \in \mathbb{H}_{\boldsymbol{\Sigma}}$, and the weak modes of $\mathrm{P}_{\mathbb{B}}^{\boldsymbol{Y},\lambda}$ coincide with the minimizers of*

$$R_{FSP}^\lambda: \mathbb{H}_{\boldsymbol{\Sigma}} \to \mathbb{R}, \boldsymbol{f} \mapsto \Phi^{\boldsymbol{Y},\lambda}(\boldsymbol{f}) + \frac{1}{2}\|\boldsymbol{f} - \boldsymbol{\mu}\|_{\mathbb{H}_{\boldsymbol{\Sigma}}}^2. \tag{3.4}$$

As $\lambda \to 0$, the term $\frac{1}{2\lambda^2} d_{\mathbb{B}}^2(\boldsymbol{f}, \mathbb{F})$ forces the minimizers of $R_{\mathrm{FSP}}^\lambda$ to converge to functions in $\mathbb{H}_{\boldsymbol{\Sigma}} \cap \mathbb{F}$ that minimize $R_{\mathrm{FSP}}$:

**Proposition 2.** *Let Assumptions A.1 to A.3 hold. Let $\{\lambda_n\}_{n \in \mathbb{N}} \subset \mathbb{R}_{>0}$ with $\lambda_n \to 0$, and $\{\boldsymbol{f}_n^\star\}_{n \in \mathbb{N}} \subset \mathbb{H}_{\boldsymbol{\Sigma}}$ such that $\boldsymbol{f}_n^\star$ is a minimizer of $R_{FSP}^{\lambda_n}$. Then $\{\boldsymbol{f}_n^\star\}_{n \in \mathbb{N}}$ has an $\mathbb{H}_{\boldsymbol{\Sigma}}$-weakly convergent subsequence with limit $\boldsymbol{f}^\star \in \mathbb{H}_{\boldsymbol{\Sigma}} \cap \mathbb{F}$. Moreover, $\boldsymbol{f}^\star$ is a minimizer of $R_{FSP}$ on $\mathbb{H}_{\boldsymbol{\Sigma}} \cap \mathbb{F}$.*

We prove Propositions 1 and 2 in Appendix A.2. To provide some intuition about the mode of convergence in Proposition 2, we point out that $\mathbb{H}_{\boldsymbol{\Sigma}}$-weak convergence of the subsequence $\{\boldsymbol{f}_{n_k}^\star\}_{k \in \mathbb{N}}$ implies both (strong/norm) convergence in the path space $\mathbb{B}$ of the Gaussian process $\mathbf{f}$ (i.e., $\lim_{k\to\infty} \|\boldsymbol{f}_{n_k}^\star - \boldsymbol{f}^\star\|_{\mathbb{B}} = 0$) and pointwise convergence (i.e., $\lim_{k\to\infty} \boldsymbol{f}_{n_k}^\star(\boldsymbol{x}) = \boldsymbol{f}^\star(\boldsymbol{x}) \; \forall \boldsymbol{x} \in \mathbb{X}$).

Finally, under certain technical assumptions, Theorem 4.4 from Cockayne et al. [22] can be used to show that, as $\lambda \to 0$, $\mathrm{P}_{\mathbb{B}}^{\boldsymbol{Y},\lambda}$ converges[2] to $\mathrm{P}_{\mathbb{F}}^{\boldsymbol{Y}}$ defined by the Radon-Nikodym derivative $\mathrm{P}_{\mathbb{F}}^{\boldsymbol{Y}}(\mathrm{d}\boldsymbol{f}) \propto \exp\left(-\Phi^{\boldsymbol{Y}}(\mathrm{d}\boldsymbol{f})\right) \mathrm{P}_{\mathbb{F}}(\mathrm{d}\boldsymbol{f})$ where $\mathrm{P}_{\mathbb{F}}$ is the regular conditional probability measure $\mathrm{P}_{\mathbb{B}}(\cdot \mid \mathbf{f} \in \mathbb{F})$.

---

[2] w.r.t. an integral probability metric (IPM)

4

Summarizing informally, we address the nonexistence of a density-based MAP estimator by constructing a family $\{\boldsymbol{f}^\star_\lambda\}_{\lambda>0}$ of weak modes of the related "relaxed" posteriors $\mathrm{P}^{Y,\lambda}_{\mathbb{B}}(\mathrm{d}\boldsymbol{f}) = \mathrm{P}^Y_{\mathbb{B}}(\mathrm{d}\boldsymbol{f} \mid d_{\mathbb{B}}(\boldsymbol{f}, \mathbb{F}) + \boldsymbol{\epsilon}_\lambda = 0)$ (Proposition 1). These weak modes converge to minimizers $\boldsymbol{f}^\star$ of the optimization problem in Equation (3.3) as $\lambda \to 0$ (Proposition 2). Moreover, under certain technical assumptions, the relaxed posteriors $\mathrm{P}^{Y,\lambda}_{\mathbb{B}}$ converge to the "true" posterior $\mathrm{P}^Y_{\mathbb{F}}$ as $\lambda \to 0$. Given the above, we conjecture that the $\boldsymbol{f}^\star$ are weak modes of $\mathrm{P}^Y_{\mathbb{F}}$, and leave the proof for future work. This motivates using the objective in Equation (3.3) to find the weak mode of the posterior measure $\mathrm{P}^Y_{\mathbb{F}}$ that we wish to Laplace-approximate. The next paragraph shows how this objective becomes a valid log-density.

**The FSP-LAPLACE objective as an unnormalized log-density.** As a first step, we use Algorithm 1, discussed in detail below, to train the neural network using the objective function[3]

$$R_{\mathrm{FSP}}(\boldsymbol{w}) := R_{\mathrm{FSP}}(\boldsymbol{f}(\,\cdot\,,\boldsymbol{w})) = -\sum_{i=1}^n \log p(\boldsymbol{y}^{(i)} \mid \boldsymbol{f}(\boldsymbol{x}^{(i)}, \boldsymbol{w})) + \frac{1}{2}\|\boldsymbol{f}(\,\cdot\,,\boldsymbol{w}) - \boldsymbol{\mu}\|^2_{\mathbb{H}_{\boldsymbol{\Sigma}}}. \quad (3.5)$$

We then intuitively want to use the same objective function to compute an approximate posterior over the weights using a Laplace approximation. For this to be well-defined, $R_{\mathrm{FSP}}(\boldsymbol{w})$ needs to be a valid unnormalized negative log-density, i.e., $\boldsymbol{w} \mapsto \exp(-R_{\mathrm{FSP}}(\boldsymbol{w}))$ needs to be integrable. However, without weight-space regularization, this often fails to be the case (e.g., due to continuous symmetries in weight space). Our method works around this issue by linearizing the network locally around $\boldsymbol{w}^\star$ after training (see Equation (2.2)) and then applying a Laplace approximation to $R^{\mathrm{lin}}_{\mathrm{FSP}}(\boldsymbol{w}) := R_{\mathrm{FSP}}(\boldsymbol{f}^{\mathrm{lin}}(\,\cdot\,,\boldsymbol{w}))$. In this case, the RKHS regularizer in $R^{\mathrm{lin}}_{\mathrm{FSP}}$ can be rewritten as

$$\frac{1}{2}\|\boldsymbol{f}^{\mathrm{lin}}(\,\cdot\,,\boldsymbol{w}) - \boldsymbol{\mu}\|^2_{\mathbb{H}_{\boldsymbol{\Sigma}}} = \frac{1}{2}(\boldsymbol{w} - \boldsymbol{\mu}_{\boldsymbol{w}^\star})^\top \boldsymbol{\Sigma}^\dagger_{\boldsymbol{w}^\star}(\boldsymbol{w} - \boldsymbol{\mu}_{\boldsymbol{w}^\star}) + \mathrm{const.}, \quad (3.6)$$

where $(\boldsymbol{\Sigma}^\dagger_{\boldsymbol{w}^\star})_{ij} := \langle(\boldsymbol{J}_{\boldsymbol{w}^\star})_i, (\boldsymbol{J}_{\boldsymbol{w}^\star})_j\rangle_{\mathbb{H}_{\boldsymbol{\Sigma}}}$ (here, $\boldsymbol{\Sigma}^\dagger_{\boldsymbol{w}^\star}$ is the Moore–Penrose pseudoinverse), $\boldsymbol{v}_i := \langle(\boldsymbol{J}_{\boldsymbol{w}^\star})_i, \boldsymbol{f}(\,\cdot\,,\boldsymbol{w}^\star) - \boldsymbol{\mu}\rangle_{\mathbb{H}_{\boldsymbol{\Sigma}}}$, $\boldsymbol{\mu}_{\boldsymbol{w}^\star} := \boldsymbol{w}^\star - \boldsymbol{\Sigma}_{\boldsymbol{w}^\star}\boldsymbol{v}$. Crucially, $\boldsymbol{\Sigma}_{\boldsymbol{w}^\star}$ is positive-(semi)definite. This means that $\exp(-R^{\mathrm{lin}}_{\mathrm{FSP}}(\,\cdot\,))$ is normalizable over $\mathrm{im}(\boldsymbol{\Sigma}_{\boldsymbol{w}^\star})$, i.e., we don't integrate over the null space. Note that this approximation also induces a (potentially degenerate) Gaussian "prior" $\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{w}^\star}, \boldsymbol{\Sigma}_{\boldsymbol{w}^\star})$ over the weights.

Using model linearization, we can also establish a correspondence between the Gaussian process prior in function space and the induced prior over the weights. Namely, the resulting expression is the Lebesgue density of the $\mathbb{H}_{\boldsymbol{\Sigma}}$-orthogonal projection of the GP prior onto the finite-dimensional subspace spanned by the "feature functions" $(\boldsymbol{J}_{\boldsymbol{w}^\star})_i$ learned by the neural network. Hence, our model inherits the prior structure in function space on the features induced by the Jacobian, zeroing out the probability mass in the remaining directions.

## 3.2 Algorithmic Considerations

**Training with the FSP-LAPLACE objective function.** Evaluating the FSP-LAPLACE objective proposed in the previous section requires computing the RKHS norm of the neural network. Unfortunately, this does not admit a closed-form expression in general. Hence, we use the approximation

$$\|\boldsymbol{f}(\,\cdot\,,\boldsymbol{w}) - \boldsymbol{\mu}\|^2_{\mathbb{H}_{\boldsymbol{\Sigma}}} \approx \underbrace{(\boldsymbol{f}(\boldsymbol{C},\boldsymbol{w}) - \boldsymbol{\mu}(\boldsymbol{C}))^\top \boldsymbol{\Sigma}(\boldsymbol{C},\boldsymbol{C})^{-1}(\boldsymbol{f}(\boldsymbol{C},\boldsymbol{w}) - \boldsymbol{\mu}(\boldsymbol{C}))}_{=\|\boldsymbol{h}_{\boldsymbol{C}}(\,\cdot\,,\boldsymbol{w})\|^2_{\mathbb{H}_{\boldsymbol{\Sigma}}}}, \quad (3.7)$$

where $\boldsymbol{C} \in \mathbb{X}^{n_C}$ is a set of $n_C$ *context points* and $\boldsymbol{h}_{\boldsymbol{C}}(\boldsymbol{x},\boldsymbol{w}) := \boldsymbol{\Sigma}(\boldsymbol{x},\boldsymbol{C})\boldsymbol{\Sigma}(\boldsymbol{C},\boldsymbol{C})^{-1}(\boldsymbol{f}(\boldsymbol{C},\boldsymbol{w}) - \boldsymbol{\mu}(\boldsymbol{C})) \in \mathbb{H}_{\boldsymbol{\Sigma}}$. The function $\boldsymbol{h}_{\boldsymbol{C}}$ is the minimum-norm interpolant of $\boldsymbol{f}(\,\cdot\,,\boldsymbol{w}) - \boldsymbol{\mu}$ at $\boldsymbol{C}$ in $\mathbb{H}_{\boldsymbol{\Sigma}}$. Hence, the estimator of the RKHS norm provably underestimates, i.e., $\|\boldsymbol{h}_{\boldsymbol{C}}(\,\cdot\,,\boldsymbol{w})\|^2_{\mathbb{H}_{\boldsymbol{\Sigma}}} \leq \|\boldsymbol{f}(\,\cdot\,,\boldsymbol{w}) - \boldsymbol{\mu}\|^2_{\mathbb{H}_{\boldsymbol{\Sigma}}}$.

During training, we need to compute $\|\boldsymbol{h}_{\boldsymbol{C}}(\,\cdot\,,\boldsymbol{w})\|^2_{\mathbb{H}_{\boldsymbol{\Sigma}}}$ at every optimizer step. Since this involves solving a linear system in $n_C$ unknowns and, more importantly, evaluating the neural network on the $n_C$ context points, we need to keep $n_C$ small for computational efficiency. We find that sampling an i.i.d. set of context points at every training iteration from a distribution $\mathrm{P}_C$ is an effective strategy for keeping $n_C$ small while ensuring that the neural network is appropriately regularized. The resulting training procedure is outlined in Algorithm 1.

---

[3]In order to simplify notation, we will assume $\mathbb{F} \subset \mathbb{H}_{\boldsymbol{\Sigma}}$ for the remainder of the main paper.

---

**Algorithm 1** RKHS-regularized model training

---

1: **function** FSP-LAPLACE-TRAIN($\boldsymbol{f}, \boldsymbol{w}, \mathcal{GP}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \mathrm{P}_C, \mathbb{D}, b$)
2:    **for all** minibatch $\mathcal{B} = (\boldsymbol{X}_\mathcal{B}, \boldsymbol{Y}_\mathcal{B}) \sim \mathbb{D}$ of size $b$ **do**
3:      $R_{\mathrm{FSP}}^{(1)}(\boldsymbol{w}^{(i)}) \leftarrow -\frac{n}{b} \sum_{j=1}^{b} \log p(\boldsymbol{y}_\mathcal{B}^{(j)} \mid \boldsymbol{f}(\boldsymbol{x}_\mathcal{B}^{(j)}, \boldsymbol{w}))$        ▷ *Negative log-likelihood*
4:      $\boldsymbol{C} \leftarrow (\boldsymbol{c}^{(j)})_{j=1}^{n_C} \overset{\text{i.i.d.}}{\sim} \mathrm{P}_C$              ▷ *Sample $n_C$ context points*
5:      $R_{\mathrm{FSP}}^{(2)}(\boldsymbol{w}) \leftarrow \frac{1}{2}(\boldsymbol{f}(\boldsymbol{C}, \boldsymbol{w}) - \boldsymbol{\mu}(\boldsymbol{C}))^\top \boldsymbol{\Sigma}(\boldsymbol{C}, \boldsymbol{C})^{-1}(\boldsymbol{f}(\boldsymbol{C}, \boldsymbol{w}) - \boldsymbol{\mu}(\boldsymbol{C}))$    ▷ *Equation* (3.7)
6:      $\boldsymbol{w} \leftarrow \text{OPTIMIZERSTEP}(R_{\mathrm{FSP}}^{(1)} + R_{\mathrm{FSP}}^{(2)}, \boldsymbol{w})$
7:    **return** $\boldsymbol{w}$

---

---

**Algorithm 2** Linearized Laplace approximation with Gaussian process priors

---

1: **function** FSP-LAPLACE($\boldsymbol{f}, \mathcal{GP}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \boldsymbol{C}, \mathbb{D}, \boldsymbol{w}^\star$)
2:    $\boldsymbol{v} \leftarrow \boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{C})\mathbf{1}/\|\boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{C})\mathbf{1}\|_2$        ▷ *Initial Lanczos vector (using forward-mode autodiff)*
3:    $\boldsymbol{L} \leftarrow \text{LANCZOS}(\boldsymbol{\Sigma}(\boldsymbol{C}, \boldsymbol{C}), \boldsymbol{v}) \in \mathbb{R}^{n_C d' \times r}$     ▷ $r \ll \min(n_C d', p)$ and $\boldsymbol{L}\boldsymbol{L}^\top \approx \boldsymbol{\Sigma}(\boldsymbol{C}, \boldsymbol{C})^\dagger$
4:    $\boldsymbol{M} \leftarrow \boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{C})^\top \boldsymbol{L} \in \mathbb{R}^{p \times r}$          ▷ *using backward-mode autodiff*
5:    $(\boldsymbol{U}_M, \boldsymbol{D}_M, \dots) \leftarrow \text{svd}(\boldsymbol{M})$        ▷ $\boldsymbol{M}\boldsymbol{M}^\top = \boldsymbol{U}_M \boldsymbol{D}_M^2 \boldsymbol{U}_M^\top$ where $\boldsymbol{U}_M \in \mathbb{R}^{p \times r}$
6:    $\boldsymbol{A} \leftarrow \boldsymbol{D}_M^2 - \sum_{i=1}^{n} \boldsymbol{U}_M^\top \boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{x}^{(i)})^\top \boldsymbol{H}_{\boldsymbol{w}^\star}^{(i)} \boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{x}^{(i)}) \boldsymbol{U}_M$    ▷ *using forward-mode autodiff*
7:    $(\boldsymbol{U}_A, \boldsymbol{D}_A) \leftarrow \text{eigh}(\boldsymbol{A})$                 ▷ $\boldsymbol{U}_A, \boldsymbol{D}_A \in \mathbb{R}^{r \times r}$
8:    $\boldsymbol{S} \leftarrow \boldsymbol{U}_M \boldsymbol{U}_A \boldsymbol{D}_A^{\dagger/2} \in \mathbb{R}^{p \times r}$              ▷ $\boldsymbol{S}\boldsymbol{S}^\top \approx \boldsymbol{\Lambda}^\dagger$
9:    Find smallest $k$ such that $\text{diag}\left(\boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{C}_i)\boldsymbol{S}_{:,k:r}(\boldsymbol{S}_{:,k:r})^\top \boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{C}_i)^\top\right) \leq \text{diag}\,\boldsymbol{\Sigma}(\boldsymbol{C}_i, \boldsymbol{C}_i)$    $\forall i$
10:   **return** $\mathcal{N}\left(\boldsymbol{w}^\star, \boldsymbol{S}_{:,k:r}(\boldsymbol{S}_{:,k:r})^\top\right)$

---

**Efficient linearized Laplace approximations of the FSP-LAPLACE objective.** Once a minimum $\boldsymbol{w}^\star$ of $R_{\mathrm{FSP}}$ is found, we compute a linearized Laplace approximation at $\boldsymbol{w}^\star$. The Hessian of $R_{\mathrm{FSP}}$ is then

$$\boldsymbol{\Lambda} = \boldsymbol{\Sigma}_{\boldsymbol{w}^\star}^\dagger - \sum_{i=1}^{n} \boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{x}^{(i)})^\top \boldsymbol{H}_{\boldsymbol{w}^\star}^{(i)} \boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{x}^{(i)}). \tag{3.8}$$

Again, the RKHS inner products in the entries of $\boldsymbol{\Sigma}_{\boldsymbol{w}^\star}^\dagger$ do not admit general closed-form expressions. Hence, we use the same strategy as before to estimate $\boldsymbol{\Sigma}_{\boldsymbol{w}^\star}^\dagger \approx \boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{C})^\top \boldsymbol{\Sigma}(\boldsymbol{C}, \boldsymbol{C})^{-1} \boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{C})$ at another set $\boldsymbol{C}$ of context points. Unlike above, for a Laplace approximation, it is vital to use a large number of context points to capture the prior beliefs well. Luckily, $\boldsymbol{\Sigma}_{\boldsymbol{w}^\star}^\dagger$ only needs to be computed once, at the end of training. But for large networks and large numbers of context points, it is still infeasible to compute or even represent $\boldsymbol{\Sigma}_{\boldsymbol{w}^\star}^\dagger$ in memory. To address this problem, we devise an efficient routine for computing (a square root of) the approximate posterior covariance matrix $\boldsymbol{\Lambda}^\dagger$, outlined in Algorithm 2. Our method is *matrix-free*, i.e. it never explicitly constructs any big (i.e., $p \times p$, $p \times n_C$, or $n_C \times n_C$) matrices in memory. This allows the method to scale to large models.

We start by computing a rank $r \ll \min(n_C d', p)$ approximation of the (pseudo-)inverted kernel Gram matrix $\boldsymbol{\Sigma}(\boldsymbol{C}, \boldsymbol{C})^\dagger \approx \boldsymbol{L}\boldsymbol{L}^\top$ using fully-reorthogonalized Lanczos iteration [23, Section 10.1] with an embedded $\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^\top$-factorization [23, Section 11.3.5]. This only needs access to matrix-vector products with $\boldsymbol{\Sigma}(\boldsymbol{C}, \boldsymbol{C})$, which can be implemented efficiently without materializing the matrix in memory. Moreover, kernel Gramians typically exhibit rapid spectral decay [see e.g., 24], which makes the low-rank approximation particularly accurate. The low-rank factors $\boldsymbol{L}$ then yield a rank $r$ approximation of $\boldsymbol{\Sigma}_{\boldsymbol{w}^\star}^\dagger \approx \boldsymbol{M}\boldsymbol{M}^\top$ with $\boldsymbol{M} := \boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{C})^\top \boldsymbol{L}$, which is embarrassingly parallel and can be computed using backward-mode autodiff to avoid materializing the network Jacobians in memory. An eigendecomposition of $\boldsymbol{M}\boldsymbol{M}^\top = \boldsymbol{U}_M \boldsymbol{D}_M^2 \boldsymbol{U}_M^\top$ can be computed in $\mathcal{O}(pr^2)$ time from a thin SVD of $\boldsymbol{M} = \boldsymbol{U}_M \boldsymbol{D}_M \boldsymbol{V}_M^\top$. The eigenvectors $\boldsymbol{U}_M$ define an orthogonal projector $\boldsymbol{P} := \boldsymbol{U}_M \boldsymbol{U}_M^\top$ onto the range and an orthogonal projector $\boldsymbol{P}_0 := \boldsymbol{I} - \boldsymbol{U}_M \boldsymbol{U}_M^\top$ onto the nullspace of $\boldsymbol{M}\boldsymbol{M}^\top$. This means that we have the decomposition $\boldsymbol{\Lambda} = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^\top + \boldsymbol{P}_0\boldsymbol{\Lambda}\boldsymbol{P}_0^\top = \boldsymbol{U}_M \boldsymbol{A} \boldsymbol{U}_M^\top + \boldsymbol{P}_0\boldsymbol{\Lambda}\boldsymbol{P}_0^\top$ with

$$\boldsymbol{A} := \boldsymbol{U}_M^\top \boldsymbol{\Lambda} \boldsymbol{U}_M = \boldsymbol{D}_M^2 - \sum_{i=1}^{n} \boldsymbol{U}_M^\top \boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{x}^{(i)})^\top \boldsymbol{H}_{\boldsymbol{w}^\star}^{(i)} \boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{x}^{(i)}) \boldsymbol{U}_M. \tag{3.9}$$

We find that $\boldsymbol{P}_0 \boldsymbol{\Lambda} \boldsymbol{P}_0^\top$ is negligible in practice, and hence, we approximate $\boldsymbol{\Lambda} \approx \boldsymbol{U}_M \boldsymbol{A} \boldsymbol{U}_M^\top$ (see Table C.2).

Finally, by computing an eigendecomposition of $\boldsymbol{A} = \boldsymbol{U}_A \boldsymbol{D}_A \boldsymbol{U}_A^\top \in \mathbb{R}^{r \times r}$ in $\mathcal{O}(r^3)$ time, we obtain an eigendecomposition of $\boldsymbol{\Lambda} \approx \boldsymbol{U}_M \boldsymbol{A} \boldsymbol{U}_M^\top = (\boldsymbol{U}_M \boldsymbol{U}_A) \boldsymbol{D}_A (\boldsymbol{U}_M \boldsymbol{U}_A)^\top$, which can be used to obtain a matrix-free representation of (a square root of) the (pseudo-)inverse of $\boldsymbol{\Lambda}$. Unfortunately, due to numerical imprecision, it is difficult to distinguish between zero eigenvalues of $\boldsymbol{A}$ and those with small positive magnitudes. Since we need to pseudo-invert the matrix to obtain the covariance matrix, this can explode predictive variance. As a remedy, we use a heuristic based on the observation that, in a linear-Gaussian model, the marginal variance of the posterior is always upper-bounded by the marginal variance of the prior. Hence, we impose that $\mathrm{diag}\left(\boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{C}_i) \boldsymbol{\Lambda}^\dagger \boldsymbol{J}_{\boldsymbol{w}^\star}(\boldsymbol{C}_i)\right) \leq \mathrm{diag}\, \boldsymbol{\Sigma}(\boldsymbol{C}_i, \boldsymbol{C}_i)$ for all $i = 1, \ldots, n_C$ by successively truncating the smallest eigenvalues in $\boldsymbol{D}_A$ until the condition is fulfilled. This turns out to be an effective strategy to combat exploding predictive variance in practice.

**Choice of context points.** Methods for regularizing neural networks in function space rely on a set of context points to evaluate the regularizer [8, 9, 25, 26]. The context points should cover the set of input locations where it is plausible that the model might be evaluated when deployed. Popular strategies include uniform sampling from a bounded subset of the input space [8, 9, 25, 26], from the training data [8], and from additional (possibly unlabeled) datasets [25, 26]. For MAP estimation, we choose a uniform context point distribution $\mathrm{P}_C$ or sample from other datasets for high-dimensional inputs. We compute the posterior covariance using samples $\boldsymbol{C}$ from a low-discrepancy sequence, which effectively cover high-dimensional spaces.

## 4 Experiments

We evaluate FSP-LAPLACE on synthetic and real-world data, demonstrating that our method effectively incorporates beliefs specified through a GP prior; that it improves performance on regression tasks for which we have prior knowledge in the form of a kernel (Mauna Loa and ocean current modeling); and that it shows competitive performance on regression, classification, out-of-distribution detection, and Bayesian optimization tasks compared to baselines.

**Baselines.** We compare FSP-LAPLACE to a deterministic neural network fit using maximum a-posteriori estimation with an isotropic Gaussian prior on the weights (MAP) and a neural network for which we additionally compute the standard linearized Laplace approximation (Laplace) [17]. We further compare our method to FVI [8], which uses GP priors with VI, to a Gaussian process (GP) [13] when the size of the dataset allows it, and to a sparse Gaussian process (sparse GP) [27]. We use the full Laplace approximation when the size of the neural network allows it, and otherwise consider the K-FAC or diagonal approximations [28]. All neural networks share the same architecture.

**Qualitative evaluation on synthetic data.** We consider two synthetic data tasks: a 1-dimensional regression task with randomly drawn noisy measurements of the function $y = \sin(2\pi x)$, and the 2-dimensional two-moons classification task from the `scikit-learn` library [29]. For the regression task, data points are shown as gray circles, functions drawn from the posterior as green lines and the inferred mean function as a red line (see Figures 1 and C.1 to C.4). For the classification task, we plot the predictive mean and 2-standard-deviations of the predictions for class 1 (blue circles) in Figures C.5 and C.6. We apply FSP-LAPLACE to the data with different GP priors and find that it successfully adapts to the specified beliefs. For instance, by varying the GP prior, we can make our method generate functions that are periodic within the support of the context points (Figure C.2), and control their smoothness (Figure 1) and length scale (Figure C.3) without modifying the network's architecture, or adding features. Such flexibility is impossible with the isotropic Gaussian weight-space priors used in the Laplace and MAP baselines. These results carry over to classification, where our method shows a smooth decision boundary when equipped with an RBF kernel (Figure C.6), and a rough decision boundary when equipped with a Matern-1/2 kernel (Figure C.5). Unlike the Laplace and MAP baselines, whose predictions remain confident beyond the support of the data, the FSP-LAPLACE's posterior mean reverts to the zero-mean prior, and its posterior variance increases. Our method also captures the properties specified by the GP prior better than FVI, especially for rougher Matern-1/2 kernels (Figures 1 and C.5). While a sufficient number of context points is necessary to capture the beliefs specified by the GP prior, we find that, even with a very small number
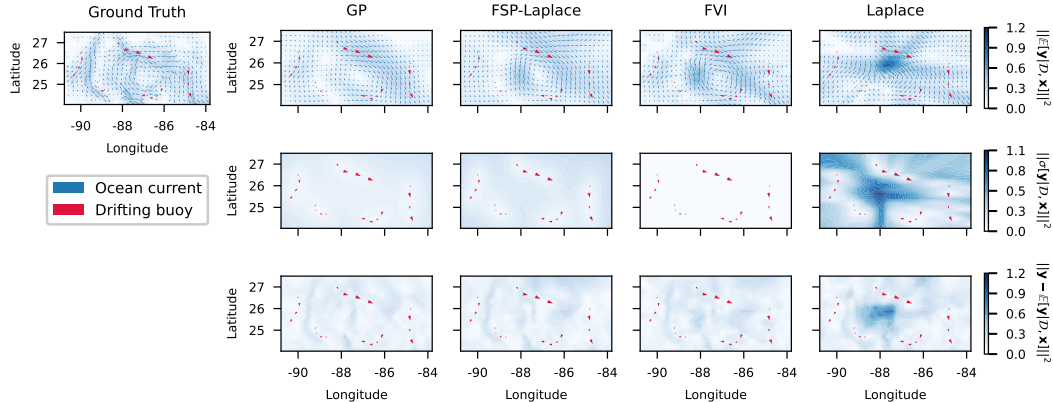
Figure 2: Results for the ocean current modeling experiment. We report the mean velocity vectors, the norm of their standard-deviation and the squared errors of compared methods. Unlike the Laplace, we find that FSP-LAPLACE accurately captures ocean current dynamics.

Table 1: Results for the Mauna Loa $CO_2$ prediction and ocean current modeling tasks. Incorporating knowledge via the GP prior in our FSP-LAPLACE improves performance over the standard Laplace.

| DATASET | EXPECTED LOG-LIKELIHOOD (↑) | | | | MEAN SQUARED ERROR (↓) | | | |
|---|---|---|---|---|---|---|---|---|
| | FSP-LAPLACE (OURS) | FVI | LAPLACE | GP | FSP-LAPLACE (OURS) | FVI | LAPLACE | GP |
| MAUNA LOA | -6'015.39 ± 443.13 | -13'594.57 ± 1'303.26 | -16'139.85 ± 2'433.70 | **-2'123.90 ± 0.00** | **8.90 ± 1.15** | 38.38 ± 2.10 | 28.26 ± 6.87 | 35.66 ± 0.00 |
| OCEAN CURRENT | **0.2823 ± 0.0009** | -35.2741 ± 1.0278 | -126'845.4063 ± 14'470.5270 | -0.5069 ± 0.0000 | 0.0169 ± 0.0001 | 0.0184 ± 0.0003 | 0.0320 ± 0.0016 | **0.0131 ± 0.0000** |

of context points, our method produces useful uncertainty estimates and reverts to the prior mean (Figures C.7 and C.8).

## 4.1 Quantitative evaluation on real-world data

We now move on to investigate FSP-LAPLACE on two real-world scientific modeling tasks: forecasting the concentration of $CO_2$ at the Mauna Loa observatory and predicting ocean currents in the Gulf of Mexico. We then assess the performance of our method on standard benchmark regression, classification, out-of-distribution detection, and Bayesian optimization tasks. When reporting results, we bold the score with the highest mean as well as any scores whose standard-error bars overlap.

**Mauna Loa dataset.** We consider the task of modeling the monthly average atmospheric $CO_2$ concentration at the Mauna Loa observatory in Hawaii from 1974 to 2024 using data collected from the NORA global monitoring laboratory[4] [30]. This 1-dimensional dataset is very accurately modeled by a combination of multiple kernels proposed in Rasmussen and Williams [13, Section 5.4.3]. We equip FSP-LAPLACE with this informative kernel and with additional periodic features, and compare to FVI with the same prior and additional periodic features, to a GP with the same prior, and to the linearized Laplace with the same additional periodic features. Using periodic features (partially) reflects the prior knowledge carried by the kernel. Results are presented by Table 1 and Figure C.9 in the Appendix. We find that incorporating prior beliefs both via an informative prior and periodic features in FSP-LAPLACE significantly reduces the mean squared error (MSE) compared to Laplace and GP baselines, and that our method is also more accurate than FVI, which uses variational inference. In terms of expected log-likelihood, all neural networks under-estimate the likelihood scale, which results in poorer scores than the GP.

**Ocean current modeling.** We further evaluate FSP-LAPLACE's ability to take into account prior knowledge by considering an ocean current modeling task where we are given sparse 2-dimensional observations of velocities of ocean drifter buoys, and we are interested in estimating ocean currents further away from the buoys. For this, we consider the GulfDrifters dataset [31] and we follow the setup by Shalashilin [32]. We incorporate known physical properties of ocean currents into the considered models by applying the Helmholtz decomposition to the GP prior's kernel [33] as well as to the neural network. Results are shown in Table 1 and in Figure 2. We find that FSP-LAPLACE

---

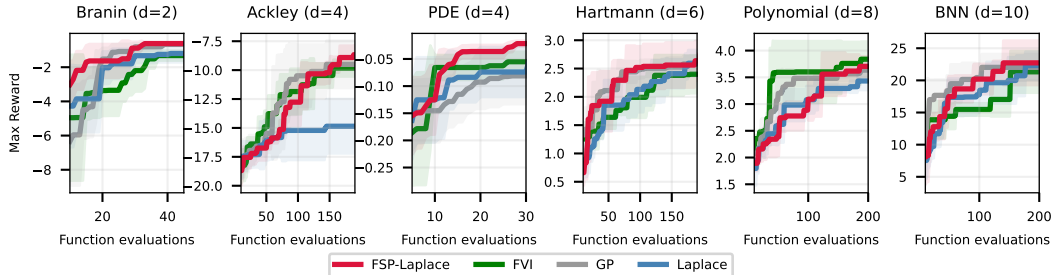[4]https://gml.noaa.gov/ccgg/trends/data.html

8

Figure 3: Results using our method (FSP-LAPLACE) as a surrogate model for Bayesian optimization. We find that FSP-LAPLACE performs particularly well on lower-dimensional problems, where it converges more quickly and to higher rewards than the Laplace, obtaining comparable scores as the Gaussian process (GP).

Table 2: Test expected log-likelihood, accuracy, expected calibration error and OOD detection accuracy on MNIST and FashionMNIST. FSP-LAPLACE performs strongly among baselines matching the accuracy of best-performing baselines and obtaining the highest expected log-likelihood and out-of-distribution detection accuracy.

| DATASET | METRIC | FSP-LAPLACE (KMNIST) | FSP-LAPLACE (RND) | FVI (KMNIST) | FVI (RND) | LAPLACE | MAP | SPARSE GP |
|---------|--------|----------------------|-------------------|--------------|-----------|---------|-----|-----------|
| MNIST | LOG-LIKELIHOOD (↑) | $-0.043 \pm 0.001$ | $\mathbf{-0.037 \pm 0.001}$ | $-0.238 \pm 0.006$ | $-0.145 \pm 0.005$ | $-0.043 \pm 0.001$ | $-0.039 \pm 0.001$ | $-0.301 \pm 0.002$ |
| | ACCURACY (↑) | $\mathbf{0.989 \pm 0.000}$ | $\mathbf{0.989 \pm 0.000}$ | $0.943 \pm 0.001$ | $0.976 \pm 0.001$ | $\mathbf{0.989 \pm 0.001}$ | $0.987 \pm 0.000$ | $0.930 \pm 0.001$ |
| | ECE (↓) | $\mathbf{0.002 \pm 0.000}$ | $\mathbf{0.002 \pm 0.000}$ | $0.073 \pm 0.003$ | $0.064 \pm 0.001$ | $0.013 \pm 0.001$ | $0.003 \pm 0.000$ | $0.035 \pm 0.001$ |
| | OOD ACCURACY (↑) | $\mathbf{0.977 \pm 0.003}$ | $0.895 \pm 0.011$ | $0.891 \pm 0.006$ | $0.894 \pm 0.010$ | $0.907 \pm 0.016$ | $0.889 \pm 0.009$ | $0.904 \pm 0.020$ |
| FMNIST | LOG-LIKELIHOOD (↑) | $\mathbf{-0.255 \pm 0.002}$ | $-0.259 \pm 0.003$ | $-0.311 \pm 0.005$ | $-0.300 \pm 0.002$ | $-0.290 \pm 0.006$ | $-0.281 \pm 0.002$ | $-0.479 \pm 0.002$ |
| | ACCURACY (↑) | $\mathbf{0.909 \pm 0.001}$ | $\mathbf{0.909 \pm 0.001}$ | $0.906 \pm 0.002$ | $\mathbf{0.910 \pm 0.002}$ | $0.897 \pm 0.002$ | $0.900 \pm 0.001$ | $0.848 \pm 0.001$ |
| | ECE (↓) | $0.018 \pm 0.002$ | $0.020 \pm 0.003$ | $0.024 \pm 0.002$ | $0.027 \pm 0.005$ | $\mathbf{0.014 \pm 0.002}$ | $0.016 \pm 0.002$ | $0.027 \pm 0.001$ |
| | OOD ACCURACY (↑) | $\mathbf{0.994 \pm 0.001}$ | $0.797 \pm 0.007$ | $0.975 \pm 0.002$ | $0.925 \pm 0.005$ | $0.801 \pm 0.014$ | $0.794 \pm 0.010$ | $0.938 \pm 0.007$ |

strongly improves over Laplace, FVI and GP in terms of expected log-likelihood, and performs competitively in terms of mean squared error (MSE). FSP-LAPLACE also improves MSE and test expected log-likelihood over FVI, which strongly underestimates the predictive variance.

**Image classification.** We further evaluate our method on the MNIST [34] and FashionMNIST [35] image classification datasets using a convolutional neural network. We compare our model to FVI, Laplace, MAP, and Sparse GP baselines, as the scale of the datasets forbids exact GP inference. For FSP-LAPLACE and FVI, we compare using context points drawn from a uniform distribution (RND) and drawn from the Kuzushiji-MNIST dataset (KMNIST) following the setup by Rudner et al. [25]. Results are presented in Table 2. Although these datasets are particularly challenging to our method, which is regularized in function space, we find that FSP-LAPLACE performs strongly and matches or exceeds the expected log-likelihood and predictive accuracy of best-performing baselines. It also yields well-calibrated models with low expected calibration error (ECE).

**Out-of-distribution detection.** We now investigate whether the epistemic uncertainty of FSP-LAPLACE is predictive for out-of-distribution detection (OOD). We follow the setup by Osawa et al. [36] and report the accuracy of a single threshold to classify OOD from in-distribution (ID) data based on the predictive uncertainty. Additional details are provided in Appendix B.2. FSP-LAPLACE with context points sampled from KMNIST performs strongly, obtaining the highest out-of-distribution detection accuracy (see Table 2, note that FSP-LAPLACE makes no assumption on whether context points are in or out of distribution). This can be further observed in Figure C.10 in the Appendix, where the predictive entropy of ID data points is tightly peaked around 0, whereas the predictive entropy of OOD data points is highly concentrated around the maximum entropy of the softmax distribution ($\ln(10) \approx 2.3$). With RND context points, our method performs comparably to the Laplace baseline.

**Bayesian optimization.** We finally evaluate the epistemic uncertainty of FSP-LAPLACE as a surrogate model for Bayesian optimization. We consider a setup derived from Li et al. [7], comparing to FVI, the linearized Laplace, and to a GP. Results are summarized in Figure 3. We find that our method performs particularly well on lower-dimensional tasks, where it converges both faster and to higher rewards than Laplace, and noticeably strongly improves over a Gaussian process on PDE. On higher-dimensional tasks, our method performs comparatively well to Laplace and GP baselines.

9

# 5   Related work

**Laplace approximation in neural networks.**   First introduced by MacKay [37], the Laplace approximation gained strong traction in the Bayesian deep learning community with the introduction of scalable log-posterior Hessian approximations [28, 38], and the so-called linearized Laplace, which solves the underfitting issue observed with standard Laplace [17, 39–41]. In addition to epistemic uncertainty estimates, the Laplace approximation also provides a method to select prior parameters via marginal likelihood estimation [11, 12]. Recent work has made the linearized Laplace more scalable by restricting inference to a subset of parameters [42], by exploiting its GP formulation [17] to apply methods from the scalable GP literature [43–45], or by directly sampling from the Laplace approximation without explicitly computing the covariance matrix [46]. While these approaches use the GP formulation to make the linearized Laplace more scalable, we are unaware of any method that uses GP priors to incorporate interpretable prior beliefs within the Laplace approximation.

**BNNs with function-space priors.**   In the context of variational inference, function-space priors in BNNs demonstrate improvements in predictive performance compared to weight-space priors [8]. While this idea might seem sound, it turns out that the KL divergence in the VI objective is infinite for most cases of interest due to mismatching supports between the function-space prior and BNN's predictive posterior [14], which therefore requires additional regularization to be well defined [8]. Due to this issue, other work [47] considers generalized VI [10] using the regularized KL divergence [48] or abandons approximating the neural network's posterior and instead uses deterministic neural networks as basis functions for Bayesian linear regression [15] or for the mean of a sparse GP [49]. In contrast, our method does not compute a divergence in function space, but only the RKHS norm under the prior's kernel, thus circumventing the issue of mismatching support. Alternatively, rather than directly placing a prior on the function generated by a BNN, researchers have investigated methods to find weight-space priors whose pushforward approximates a target function-space measure by minimizing a divergence [9, 50], using the Ridgelet transform [51], or changing the BNN's architecture [52].

**Regularizing neural networks in function space.**   Arguing that one ultimately only cares about the output function of the neural network, it has been proposed to regularize neural networks in function space, both showing that norms could be efficiently estimated and that such regularization schemes performed well in practice [26, 53–55]. Unlike FSP-LAPLACE, none of these methods allow to specify informative beliefs via a GP prior. Our method uses the same RKHS norm estimator as Chen et al. [54] (however, under a different kernel) by sampling a new batch of context points at each update step. Similarly, Rudner et al. [26] propose an empirical prior on the weights that regularizes the neural network in function space, which they use for MAP estimation or approximate posterior inference. The MAP objective resembles ours, but unlike our method, uses the kernel induced by the last layer of the neural network and includes an additional Gaussian prior on the weights.

# 6   Conclusion

We propose a method for applying the Laplace approximation to neural networks with interpretable Gaussian process priors in function space. This addresses the issue that conventional applications of approximate Bayesian inference methods to neural networks require posing a prior in weight space, which is virtually impossible because weight space is not interpretable. We address the non-existence of densities in (infinite-dimensional) function space by generalizing the notion of a MAP estimate to the limit of a sequence of weak modes of related posterior measures, leading us to propose a simple objective function. We further mitigate the computational cost of calculating high-dimensional curvature matrices using scalable methods from matrix-free linear algebra. By design, our method works best in application domains where prior information can be encoded in the language of Gaussian processes. This is confirmed by experiments on scientific data and Bayesian optimization. In high-dimensional spaces, where explicit prior knowledge is difficult to state, Gaussian process priors are naturally at a disadvantage. While we do demonstrate superior performance on image data, it is yet unclear how to find good function-space priors in such high-dimensional spaces.

## Acknowledgments and Disclosure of Funding

## Bibliography

[1] Nina Effenberger, Marvin Pförtner, Philipp Hennig, and Nicole Ludwig. Probabilistic wind speed downscaling for future wind power assessment. URL https://meetingorganizer.copernicus.org/EGU24/EGU24-2034.html.

[2] Abhaya Indrayan. Aleatory and epistemic uncertainties can completely derail medical research results. *Journal of postgraduate medicine*, 66, 03 2020. doi:10.4103/jpgm.JPGM_585_19.

[3] Kenza Tazi, Jihao Andreas Lin, Alex S Gardner, ST John, Hong Ge, and Richard E Turner. Towards more interpretable and robust geospatial modelling with gaussian processes. *AGU23*, 2023.

[4] Alexander Immer, Maciej Korzepa, and Matthias Bauer. Improving predictions of bayesian neural nets via local linearization, 2021.

[5] Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux – effortless bayesian deep learning, 2022.

[6] Vincent Fortuin. Priors in bayesian deep learning: A review, 2022.

[7] Yucen Lily Li, Tim G. J. Rudner, and Andrew Gordon Wilson. A study of bayesian neural network surrogates for bayesian optimization, 2024.

[8] Shengyang Sun, Guodong Zhang, Jiaxin Shi, and Roger Grosse. Functional variational bayesian neural networks, 2019.

[9] Ba-Hien Tran, Simone Rossi, Dimitrios Milios, and Maurizio Filippone. All you need is a good functional prior for bayesian deep learning, 2022.

[10] Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. Generalized variational inference: Three arguments for deriving new posteriors, 2019.

[11] Alexander Immer, Matthias Bauer, Vincent Fortuin, Gunnar Rätsch, and Mohammad Emtiyaz Khan. Scalable marginal likelihood estimation for model selection in deep learning, 2021.

[12] Javier Antorán, David Janz, James Urquhart Allingham, Erik Daxberger, Riccardo Barbano, Eric Nalisnick, and José Miguel Hernández-Lobato. Adapting the linearised laplace model evidence for modern deep learning, 2022.

[13] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning.* Adaptive computation and machine learning. MIT Press, 2006. ISBN 026218253X.

[14] David R. Burt, Sebastian W. Ober, Adrià Garriga-Alonso, and Mark van der Wilk. Understanding variational inference in function-space, 2020.

[15] Chao Ma and José Miguel Hernández-Lobato. Functional variational inference based on stochastic process generators. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 21795–21807. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/b613e70fd9f59310cf0a8d33de3f2800-Paper.pdf.

[16] David J. C. MacKay. The Evidence Framework Applied to Classification Networks. *Neural Computation*, 4(5):720–736, 09 1992. ISSN 0899-7667. doi:10.1162/neco.1992.4.5.720. URL https://doi.org/10.1162/neco.1992.4.5.720.

[17] Alexander Immer, Maciej Korzepa, and Matthias Bauer. Improving predictions of bayesian neural nets via local linearization, 2021.

[18] John C. Oxtoby. Invariant measures in groups which are not locally compact. *Transactions of the American Mathematical Society*, 60:215–237, 1946. doi:10.1090/S0002-9947-1946-0018188-5.

[19] Marvin Pförtner, Ingo Steinwart, Philipp Hennig, and Jonathan Wenger. Physics-informed Gaussian process regression generalizes linear PDE solvers, 2022.

[20] Hefin Lambley. Strong maximum a posteriori estimation in Banach spaces with Gaussian priors. *Inverse Problems*, 39(12):125010, 2023. doi:10.1088/1361-6420/ad07a4.

[21] Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M. Stuart. Solving and learning nonlinear PDEs with Gaussian processes. *Journal of Computational Physics*, 447, 2021. doi:10.1016/j.jcp.2021.110668.

[22] Jon Cockayne, Chris J. Oates, T. J. Sullivan, and Mark Girolami. Bayesian probabilistic numerical methods. *SIAM Review*, 61(4):756–789, 2019. doi:10.1137/17M1139357.

[23] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. The Johns Hopkins University Press, Baltimore, fourth edition, 2013.

[24] Francis Bach. Unraveling spectral properties of kernel matrices – I, 2024. URL https://francisbach.com/spectrum-kernel-matrices-i/.

[25] Tim G. J. Rudner, Zonghao Chen, Yee Whye Teh, and Yarin Gal. Tractable function-space variational inference in bayesian neural networks, 2023.

[26] Tim G. J. Rudner, Sanyam Kapoor, Shikai Qiu, and Andrew Gordon Wilson. Function-space regularization in neural networks: A probabilistic perspective, 2023.

[27] James Hensman, Nicolo Fusi, and Neil D. Lawrence. Gaussian processes for big data, 2013.

[28] Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Skdvd2xAZ.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[30] Charles D Keeling and T P Whorf. Monthly carbon dioxide measurements on Mauna Loa, Hawaii from 1958 to 1998, 2000. URL https://doi.org/10.1594/PANGAEA.56536.

[31] Jonathan M. Lilly and Paula Pérez-Brunius. GulfDrifters: A consolidated surface drifter dataset for the Gulf of Mexico, January 2021. URL https://doi.org/10.5281/zenodo.4421585.

[32] Ivan Shalashilin. Gaussian processes for vector fields and ocean current modelling, March 2024. URL https://docs.jaxgaussianprocesses.com/examples/oceanmodelling/.

[33] Renato Berlinghieri, Brian L. Trippe, David R. Burt, Ryan Giordano, Kaushik Srinivasan, Tamay Özgökmen, Junfei Xia, and Tamara Broderick. Gaussian processes at the helm(holtz): A more fluid model for ocean currents, 2023.

[34] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

[35] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[36] Kazuki Osawa, Siddharth Swaroop, Anirudh Jain, Runa Eschenhagen, Richard E. Turner, Rio Yokota, and Mohammad Emtiyaz Khan. Practical deep learning with bayesian principles, 2019.

[37] David John Cameron MacKay. Bayesian methods for adaptive models. 1992. URL https://api.semanticscholar.org/CorpusID:123141880.

[38] James Martens. New insights and perspectives on the natural gradient method, 2020.

[39] Mohammad Emtiyaz Khan, Alexander Immer, Ehsan Abedi, and Maciej Korzepa. Approximate inference turns deep networks into gaussian processes, 2020. URL https://arxiv.org/abs/1906.01930.

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

[41] Mohammad Emtiyaz Khan and Siddharth Swaroop. Knowledge-adaptation priors, 2021. URL https://arxiv.org/abs/2106.08769.

[42] Erik Daxberger, Eric Nalisnick, James Urquhart Allingham, Javier Antorán, and José Miguel Hernández-Lobato. Bayesian deep learning via subnetwork inference, 2022.

[43] Zhijie Deng, Feng Zhou, and Jun Zhu. Accelerated linearized laplace approximation for bayesian deep learning, 2022.

[44] Luis A. Ortega, Simón Rodríguez Santana, and Daniel Hernández-Lobato. Variational linearized laplace approximation for bayesian deep learning, 2024.

[45] Aidan Scannell, Riccardo Mereu, Paul Edmund Chang, Ella Tamir, Joni Pajarinen, and Arno Solin. Function-space parameterization of neural networks for sequential learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=2dhxxIKhqz.

[46] Javier Antorán, Shreyas Padhy, Riccardo Barbano, Eric Nalisnick, David Janz, and José Miguel Hernández-Lobato. Sampling-based inference for large linear models, with application to linearised laplace, 2023.

[47] Tristan Cinquin and Robert Bamler. Regularized kl-divergence for well-defined function-space variational inference in bayesian neural networks, 2024. URL https://arxiv.org/abs/2406.04317.

[48] Minh Ha Quang. Regularized divergences between covariance operators and gaussian measures on hilbert spaces, 2019.

[49] Veit D. Wild, Robert Hu, and Dino Sejdinovic. Generalized variational inference in function spaces: Gaussian measures meet bayesian deep learning, 2022.

[50] Daniel Flam-Shepherd. Mapping gaussian process priors to bayesian neural networks. 2017. URL https://api.semanticscholar.org/CorpusID:160026812.

[51] Takuo Matsubara, Chris J. Oates, and François-Xavier Briol. The ridgelet prior: A covariance function approach to prior specification for bayesian neural networks, 2022.

[52] Tim Pearce, Russell Tsuchida, Mohamed Zaki, Alexandra Brintrup, and Andy Neely. Expressive priors in bayesian neural networks: Kernel combinations and periodic functions, 2019.

[53] Ari S. Benjamin, David Rolnick, and Konrad Kording. Measuring and regularizing networks in function space, 2019.

[54] Zonghao Chen, Xupeng Shi, Tim G. J. Rudner, Qixuan Feng, WEIZHONG ZHANG, and Tong Zhang. A neural tangent kernel perspective on function-space regularization in neural networks. In *OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop)*, 2022. URL https://openreview.net/forum?id=E6MGIXQlKw.

[55] Shikai Qiu, Tim G. J. Rudner, Sanyam Kapoor, and Andrew Gordon Wilson. Should we learn most likely functions or parameters?, 2023.

[56] Nathaël Da Costa, Marvin Pförtner, Lancelot Da Costa, and Philipp Hennig. Sample path regularity of Gaussian processes from the covariance kernel, 2023.

[57] Motonobu Kanagawa, Philipp Hennig, Dino Sejdinovic, and Bharath K. Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences, 2018.

[58] M. Dashti, K. J. H. Law, A. M. Stuart, and J. Voss. MAP estimators and their consistency in Bayesian nonparametric inverse problems. *Inverse Problems*, 29(9), September 2013. doi:10.1088/0266-5611/29/9/095017.

[59] John B. Conway. *A Course in Functional Analysis*, volume 96 of *Graduate Texts in Mathematics*. Springer, New York, NY, second edition, 1997. doi:10.1007/978-1-4757-4383-8.

[60] Vladimir Igorevich Bogachev. *Gaussian Measures*, volume 62 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, Rhode Island, 1998.

[61] Dimitrios Milios, Raffaello Camoriano, Pietro Michiardi, Lorenzo Rosasco, and Maurizio Filippone. Dirichlet-based gaussian processes for large-scale calibrated classification. In

S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/b6617980ce90f637e68c3ebe8b9be745-Paper.pdf.

[62] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Yamamoto Kazuaki, and David Ha. Deep learning for classical japanese literature, 12 2018.

[63] J.H. HALTON. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2:84–90, 1960. URL http://eudml.org/doc/131448.

[64] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[65] Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com.

[66] Hao Chen, Lili Zheng, Raed Al Kontar, and Garvesh Raskutti. Gaussian process inference using mini-batch stochastic gradient descent: Convergence guarantees and empirical benefits, 2021.

[67] M. Mckay, Richard Beckman, and William Conover. A comparison of three methods for selecting vales of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245, 05 1979. doi:10.1080/00401706.1979.10489755.

[68] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*, 2020. URL http://arxiv.org/abs/1910.06403.

[69] Dheeru Dua and Casey Graff. Uci machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

[70] Andrey Malinin, Liudmila Prokhorenkova, and Aleksei Ustimenko. Uncertainty in gradient boosting via ensembles, 2021.

[71] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[72] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

[73] Tom Hennigan, Trevor Cai, Tamara Norman, Lena Martens, and Igor Babuschkin. Haiku: Sonnet for JAX, 2020. URL http://github.com/deepmind/dm-haiku.

[74] Aleksandar Botev and James Martens. KFAC-JAX, 2022. URL https://github.com/google-deepmind/kfac-jax.

[75] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty, 2018. URL https://arxiv.org/abs/1806.01768.

# Appendix

## A Theory

### A.1 Assumptions and their applicability

**Assumption A.1.** $\mathbf{f} \sim \mathcal{GP}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a $d'$-output Gaussian process with index set $\mathbb{X}$ on $(\Omega, \mathcal{A}, \mathrm{P})$ such that

    (i) the paths of $\mathbf{f}$ lie (P-almost surely) in a real separable Banach space $\mathbb{B}$ of $\mathbb{R}^{d'}$-valued functions on $\mathbb{X}$ with continuous point evaluation maps $\delta_{\boldsymbol{x}} \colon \mathbb{B} \to \mathbb{R}^{d'}$, and

    (ii) $\omega \mapsto \mathbf{f}(\cdot, \omega)$ is a Gaussian random variable with values in $(\mathbb{B}, \mathcal{B}(\mathbb{B}))$.

We denote the law of $\omega \mapsto \mathbf{f}(\cdot, \omega)$ by $\mathrm{P}_{\mathbb{B}}$.

For this paper, we focus on $\mathbb{B} = C(\mathbb{X})$, with $\mathbb{X}$ being a compact metric space. In this case, Assumption A.1(i) can be verified from regularity properties of the prior covariance function $\boldsymbol{\Sigma}$ [see, e.g., 56]. Moreover, the sufficient criteria from Pförtner et al. [19, Section B.2] show that Assumption A.1(ii) also holds in this case.

**Assumption A.2.** The potential $\Phi^{\boldsymbol{Y}} \colon \mathbb{B} \to \mathbb{R}$ is (norm-)continuous and, for each $\eta > 0$, there is $K(\eta) \in \mathbb{R}$ such that $\Phi^{\boldsymbol{Y}}(\boldsymbol{f}) \geq K(\eta) - \eta \|\boldsymbol{f}\|_{\mathbb{B}}^2$ for all $\boldsymbol{f} \in \mathbb{B}$.

This holds if the negative log-likelihood functions $\ell^{(i)} \colon \mathbb{R}^{d'} \to \mathbb{R}$, $\boldsymbol{f}^{(i)} \mapsto -\log p(\boldsymbol{y}^{(i)} \mid \boldsymbol{f}^{(i)})$ are continuous and, for all $\eta > 0$, there is $K(\eta) \in \mathbb{R}$ such that $\ell^{(i)}(\boldsymbol{f}^{(i)}) > K(\boldsymbol{f}^{(i)}) + \eta \|\boldsymbol{f}^{(i)}\|^2$ for all $\boldsymbol{f}^{(i)} \in \mathbb{R}^{d'}$. For instance, this is true for a Gaussian likelihood $\ell^{(i)}(\boldsymbol{f}^{(i)}) = \frac{1}{2\lambda_i^2} \|\boldsymbol{y}^{(i)} - \boldsymbol{f}^{(i)}\|_2^2$.

**Assumption A.3.** (i) $\mathbb{H}_{\boldsymbol{\Sigma}} \cap \mathbb{F}$ is nonempty, (ii) $\mathbb{H}_{\boldsymbol{\Sigma}} \cap \mathbb{F}$ is closed in $\mathbb{B} \supset \mathbb{H}_{\boldsymbol{\Sigma}}$, and (iii) $\mathbb{H}_{\boldsymbol{\Sigma}} \cap \mathbb{F} \subset \mathbb{B}$ has the Heine-Borel property, i.e., all closed and bounded subsets of $\mathbb{H}_{\boldsymbol{\Sigma}} \cap \mathbb{F}$ are compact in $\mathbb{B}$.

Assumption A.3(i) can be verified using a plethora of results from RKHS theory. For instance, for Sobolev kernels like the Matérn family used in the experiments, the RKHS is norm-equivalent to a Sobolev space [see, e.g., 57]. In this case, we only need the NN to be sufficiently often (weakly) differentiable on the interior of its compact domain $\mathbb{X}$. The closure property from Assumption A.3(ii) is more difficult to verify directly. However, if we assume that $\mathbb{W}$ is compact and that the map $\mathbb{W} \to \mathbb{B}$, $\boldsymbol{w} \mapsto \boldsymbol{f}(\cdot, \boldsymbol{w})$ is continuous, then $\mathbb{F}$ is compact (and hence closed) as the image of a compact set under a continuous function. This is a reasonable assumption, since, in practice, the weights of a neural network are represented as machine numbers with a maximal magnitude, meaning that $\mathbb{W}$ is always contained in an $\ell_{\infty}$ ball of fixed radius. Incidentally, compactness of $\mathbb{F}$ also entails the Heine-Borel property from Assumption A.3(iii). Alternatively, $\mathbb{F}$ also has the Heine-Borel property if it is a topological manifold (e.g., a Banach manifold), since it is necessarily finite-dimensional.

### A.2 Proofs

**Lemma A.1.** *Let* $(\mathbb{X}, d)$ *be a metric space,* $\mathbb{A} \subseteq \mathbb{X}$ *nonempty, and*

$$d(\cdot, \mathbb{A}) \colon \mathbb{X} \to \mathbb{R}_{\geq 0}, x \mapsto \inf_{a \in \mathbb{A}} d(x, a).$$

*Then* $d(\cdot, \mathbb{A})$ *is 1-Lipschitz.*

*Proof.* For all $x_1, x_2 \in \mathbb{X}$ we have

$$
\begin{aligned}
d(x_2, \mathbb{A}) &= \inf_{a \in \mathbb{A}} d(x_2, a) \\
&\leq d(x_2, x_1) + \inf_{a \in \mathbb{A}} d(x_1, a) \\
&= d(x_1, x_2) + d(x_1, \mathbb{A})
\end{aligned}
$$

by the triangle inequality and hence $d(x_2, \mathbb{A}) - d(x_1, \mathbb{A}) \leq d(x_1, x_2)$. Since this argument is symmetric in $x_1$ and $x_2$, this also shows that

$$-(d(x_2, \mathbb{A}) - d(x_1, \mathbb{A})) = d(x_1, \mathbb{A}) - d(x_2, \mathbb{A}) \leq d(x_2, x_1) = d(x_1, x_2).$$

All in all, we obtain $|d(x_2, \mathbb{A}) - d(x_1, \mathbb{A})| \leq d(x_1, x_2)$. $\qquad \square$

**Lemma A.2.** *Let $(\mathbb{X}, d)$ be a metric space and $\mathbb{A} \subseteq \mathbb{X}$ a closed, nonempty subset with the Heine-Borel property. Then $\inf_{a \in \mathbb{A}} d(x, a)$ is attained for all $x \in \mathbb{X}$.*

*Proof.* Let $x \in \mathbb{X}$ and $r > r_x := \inf_{a \in \mathbb{A}} d(x, a)$. Then $\mathbb{A} \cap \bar{B}_r(x) \neq \emptyset$ as well as $d(x, a) > r$ for all $a \in \mathbb{A} \setminus \bar{B}_r(x)$ and thus $\inf_{a \in \mathbb{A}} d(x, a) = \inf_{a \in \mathbb{A} \cap \bar{B}_r(x)} d(x, a)$. Moreover, $\mathbb{A} \cap \bar{B}_r(x)$ is compact by the Heine-Borel property and $d(x, \cdot)$ is continuous. Hence, the claim follows from the Weierstrass extreme value theorem. $\square$

**Proposition 1.** *Let Assumptions A.1 to A.3 hold. For $\lambda > 0$, define $\Phi^{\boldsymbol{Y}, \lambda} \colon \mathbb{B} \to \mathbb{R}$, $\boldsymbol{f} \mapsto \Phi^{\boldsymbol{Y}}(\boldsymbol{f}) + \frac{1}{2\lambda^2} d_{\mathbb{B}}^2(\boldsymbol{f}, \mathbb{F})$. Then the posterior measure $\mathrm{P}_{\mathbb{B}}^{\boldsymbol{Y}, \lambda}(\mathrm{d}\boldsymbol{f}) \propto \exp\left(-\Phi^{\boldsymbol{Y}, \lambda}(\mathrm{d}\boldsymbol{f})\right) \mathrm{P}_{\mathbb{B}}(\mathrm{d}\boldsymbol{f})$ has at least one weak mode $\boldsymbol{f}^\star \in \mathbb{H}_{\boldsymbol{\Sigma}}$, and the weak modes of $\mathrm{P}_{\mathbb{B}}^{\boldsymbol{Y}, \lambda}$ coincide with the minimizers of*

$$R_{FSP}^\lambda \colon \mathbb{H}_{\boldsymbol{\Sigma}} \to \mathbb{R}, \boldsymbol{f} \mapsto \Phi^{\boldsymbol{Y}, \lambda}(\boldsymbol{f}) + \frac{1}{2}\|\boldsymbol{f} - \boldsymbol{\mu}\|_{\mathbb{H}_{\boldsymbol{\Sigma}}}^2. \tag{3.4}$$

*Proof.* $d_{\mathbb{B}}(\cdot, \mathbb{F})$ is (globally) 1-Lipschitz by Lemma A.1 and bounded from below by 0. Hence, $\Phi^{\boldsymbol{Y}, \lambda} = \Phi^{\boldsymbol{Y}} + \frac{1}{2\lambda^2} d_{\mathbb{B}}^2(\cdot, \mathbb{F})$ is continuous and for all $\eta > 0$, there is $K(\eta) \in \mathbb{R}$ such that

$$\Phi^{\boldsymbol{Y}, \lambda}(\boldsymbol{f}) \geq K(\eta) - \eta\|\boldsymbol{f}\|_{\mathbb{B}}^2 + \underbrace{\frac{1}{2\lambda^2} d_{\mathbb{B}}^2(\boldsymbol{f}, \mathbb{F})}_{\geq 0} \geq K(\eta) - \eta\|\boldsymbol{f}\|_{\mathbb{B}}^2$$

by Assumption A.2. The statement then follows from Theorem 1.1 in Lambley [20]. $\square$

**Proposition 2.** *Let Assumptions A.1 to A.3 hold. Let $\{\lambda_n\}_{n \in \mathbb{N}} \subset \mathbb{R}_{>0}$ with $\lambda_n \to 0$, and $\{\boldsymbol{f}_n^\star\}_{n \in \mathbb{N}} \subset \mathbb{H}_{\boldsymbol{\Sigma}}$ such that $\boldsymbol{f}_n^\star$ is a minimizer of $R_{FSP}^{\lambda_n}$. Then $\{\boldsymbol{f}_n^\star\}_{n \in \mathbb{N}}$ has an $\mathbb{H}_{\boldsymbol{\Sigma}}$-weakly convergent subsequence with limit $\boldsymbol{f}^\star \in \mathbb{H}_{\boldsymbol{\Sigma}} \cap \mathbb{F}$. Moreover, $\boldsymbol{f}^\star$ is a minimizer of $R_{FSP}$ on $\mathbb{H}_{\boldsymbol{\Sigma}} \cap \mathbb{F}$.*

Our proof makes use of ideas from Dashti et al. [58] and Lambley [20].

*Proof.* Without loss of generality, we assume $\boldsymbol{\mu} = \boldsymbol{0}$. By Assumption A.2, there are constants $K, \alpha > 0$ such that $R_{\mathrm{FSP}}(\boldsymbol{f}) \geq K + \alpha\|\boldsymbol{f}\|_{\mathbb{H}_{\boldsymbol{\Sigma}}}^2$ for all $\boldsymbol{f} \in \mathbb{H}_{\boldsymbol{\Sigma}}$ [20, Section 4.1]. Now fix an arbitrary $\boldsymbol{f} \in \mathbb{H}_{\boldsymbol{\Sigma}} \cap \mathbb{F}$. Then

$$\begin{aligned}
R_{\mathrm{FSP}}(\boldsymbol{f}) &= R_{\mathrm{FSP}}^{\lambda_n}(\boldsymbol{f}) \\
&\geq R_{\mathrm{FSP}}^{\lambda_n}(\boldsymbol{f}_n^\star) \\
&= R_{\mathrm{FSP}}(\boldsymbol{f}_n^\star) + \frac{1}{2\lambda_n^2} d_{\mathbb{B}}^2(\boldsymbol{f}_n^\star, \mathbb{F}) \tag{A.1} \\
&\geq K + \alpha\|\boldsymbol{f}_n^\star\|_{\mathbb{H}_{\boldsymbol{\Sigma}}}^2 + \frac{1}{2\lambda_n^2} d_{\mathbb{B}}^2(\boldsymbol{f}_n^\star, \mathbb{F}) \tag{A.2} \\
&\geq K + \alpha\|\boldsymbol{f}_n^\star\|_{\mathbb{H}_{\boldsymbol{\Sigma}}}^2,
\end{aligned}$$

and hence

$$\|\boldsymbol{f}_n^\star\|_{\mathbb{H}_{\boldsymbol{\Sigma}}}^2 \leq \frac{1}{\alpha}\left(R_{\mathrm{FSP}}(\boldsymbol{f}) - K\right),$$

i.e. the sequence $\{\boldsymbol{f}_n^\star\}_{n \in \mathbb{N}} \subset \mathbb{H}_{\boldsymbol{\Sigma}}$ is bounded. By the Banach-Alaoglu theorem [59, Theorems V.3.1 and V.4.2(d)] and the Eberlein-Šmulian theorem [59, Theorem V.13.1], there is a weakly convergent subsequence $\{\boldsymbol{f}_{n_k}^\star\}_{k \in \mathbb{N}}$ with limit $\boldsymbol{f}^\star \in \mathbb{H}_{\boldsymbol{\Sigma}}$.

We need to show that $\boldsymbol{f}^\star \in \mathbb{F}$. From Equation (A.2), it follows that

$$0 \leq d_{\mathbb{B}}(\boldsymbol{f}_{n_k}^\star, \mathbb{F}) \leq \lambda_{n_k}\sqrt{2(R_{\mathrm{FSP}}(\boldsymbol{f}) - K - \alpha\|\boldsymbol{f}_{n_k}^\star\|_{\mathbb{H}_{\boldsymbol{\Sigma}}}^2)} \leq \lambda_{n_k}\sqrt{2(R_{\mathrm{FSP}}(\boldsymbol{f}) - K)},$$

where the right-hand side converges to 0 as $k \to \infty$. Hence, $\lim_{k \to \infty} d_{\mathbb{B}}(\boldsymbol{f}_{n_k}^\star, \mathbb{F}) = 0$. The embedding $\iota \colon \mathbb{H}_{\boldsymbol{\Sigma}} \to \mathbb{B}$ is compact [60, Corollary 3.2.4] and, by Lemma A.1, $d_{\mathbb{B}}(\cdot, \mathbb{F}) \colon \mathbb{B} \to \mathbb{R}$ is continuous, which implies that $d_{\mathbb{B}}(\iota[\cdot], \mathbb{F}) \colon \mathbb{H}_{\boldsymbol{\Sigma}} \to \mathbb{R}$ is sequentially weakly continuous. Hence, $d_{\mathbb{B}}(\boldsymbol{f}^\star, \mathbb{F}) = \lim_{k \to \infty} d_{\mathbb{B}}(\boldsymbol{f}_{n_k}^\star, \mathbb{F}) = 0$ and, by Assumption A.3 and Lemma A.2, it follows that $\boldsymbol{f}^\star \in \mathbb{F}$.

16

Finally, we show that $\boldsymbol{f}^\star$ is a minimizer of $R_{\mathrm{FSP}}$ on $\mathbb{H}_{\boldsymbol{\Sigma}} \cap \mathbb{F}$. Since $\Phi^{\boldsymbol{Y}} \colon \mathbb{B} \to \mathbb{R}$ is continuous and $\iota$ is compact, $\Phi^{\boldsymbol{Y}} \circ \iota \colon \mathbb{H}_{\boldsymbol{\Sigma}} \to \mathbb{R}$ is sequentially weakly continuous. Moreover, we have $\limsup_{k \to \infty} \|\boldsymbol{f}^\star_{n_k}\|^2_{\mathbb{H}_{\boldsymbol{\Sigma}}} \geq \|\boldsymbol{f}^\star\|^2_{\mathbb{H}_{\boldsymbol{\Sigma}}}$, since Hilbert norms are sequentially weakly lower-semicontinuous. Hence, $\limsup_{k \to \infty} R_{\mathrm{FSP}}(\boldsymbol{f}^\star_{n_k}) \geq R_{\mathrm{FSP}}(\boldsymbol{f}^\star)$. Now, by Equation (A.1),

$$R_{\mathrm{FSP}}(\boldsymbol{f}) \geq R_{\mathrm{FSP}}(\boldsymbol{f}^\star) + \limsup_{k \to \infty} \frac{1}{2\lambda^2_{n_k}} d^2_{\mathbb{B}}(\boldsymbol{f}^\star_{n_k}, \mathbb{F}),$$

for all $\boldsymbol{f} \in \mathbb{H}_{\boldsymbol{\Sigma}} \cap \mathbb{F}$. For $\boldsymbol{f} = \boldsymbol{f}^\star$ this implies that $\limsup_{k \to \infty} \frac{1}{2\lambda^2_{n_k}} d^2_{\mathbb{B}}(\boldsymbol{f}^\star_{n_k}, \mathbb{F}) = 0$. All in all, we arrive at $R_{\mathrm{FSP}}(\boldsymbol{f}) \geq R_{\mathrm{FSP}}(\boldsymbol{f}^\star)$ for all $\boldsymbol{f} \in \mathbb{H}_{\boldsymbol{\Sigma}} \cap \mathbb{F}$, i.e. $\boldsymbol{f}^\star$ is a minimizer of $R_{\mathrm{FSP}}$. $\qquad\square$

## B   Experimental setup

### B.1   Qualitative experiments with synthetic data

**Regression.**   We sample points from the corresponding generative model

$$y_i = \sin(2\pi x_i) + \epsilon \quad \text{with} \quad \epsilon \sim \mathcal{N}\left(0, \sigma^2_n\right) \tag{B.1}$$

using $\sigma_n = 0.1$ and draw $x_i \sim \mathcal{U}([-1, -0.5] \cup [0.5, 1])$. We plot data points as gray circles, functions sampled from the approximate posterior as green lines, the empirical mean function as a red line and its empirical 2-standard-deviation interval around the mean as a green surface. All neural networks have the same two hidden-layer architecture with 50 neurons per layer and hyperbolic tangent ($\mathtt{tanh}$) activation functions. For FSP-LAPLACE, we use 100 context points placed on a regular grid and run a maximum of 500 Lanczos iterations. For FVI, we sample 100 context points drawn from $\mathcal{U}([-2, 2])$ at each update. Except when stated otherwise, we consider a centered GP prior and find the parameters of the covariance function by maximizing the log-marginal likelihood [13]. For the Laplace, we use the full generalized Gauss-Newton matrix and an isotropic Gaussian prior with scale $\sigma_p = 1$. The MAP estimate uses the same prior. We find the parameters of the Gaussian process priors by maximizing the log-marginal likelihood and the parameters of the sparse GP by maximizing the evidence lower bound [13].

**Classification.**   We sample randomly perturbed data points from the two moons data [29] with noise level $\sigma_n = 0.1$. We plot the the data points from class 0 as red dots and those from class 1 as blue dots. We show the mean (upper row) and 2-standard-deviation (bottom row) of the probability that a sample $\boldsymbol{x}$ belongs to class 1 under the approximate posterior, which we estimate using $K = 100$ samples. We consider a two hidden-layer neural network with 100 neurons per layer and hyperbolic tangent activation functions. For FSP-LAPLACE, we use 100 context points placed on a regular grid over $[-3.75, 3.75] \times [-3.75, 3.75]$ and limit Lanczos to run for at most 500 iterations. For FVI, we sample 100 context points from $\mathcal{U}([-3.75, 3.75]^2)$ at each update. We consider a centered GP prior and find the parameters of the covariance function by maximizing the log-marginal likelihood [13] using the reparameterization of classifications labels into regression targets from Milios et al. [61]. For the Laplace, we use the full generalized Gauss-Newton matrix and an isotropic Gaussian prior with scale $\sigma_p = 1$. The MAP estimate uses the same prior. For the Gaussian process, we Laplace-approximate the intractable GP posterior and find the prior parameters by maximizing the log-marginal likelihood [13]. Sparse GP parameters are found by maximizing the ELBO [13].

### B.2   Quantitative experiments with real-world data

**Mauna Loa.**   We consider the Mauna Loa dataset which tracks the monthly average atmospheric $CO_2$ concentration at the Mauna Loa observatory in Hawaii from 1974 to 2024 [30]. We consider the first $70\%$ of the data in chronological order as the train set (from 1974 to 2005) and the last $30\%$ as the test set (from 2009 to 2024). We standardize the features (time) and regression targets ($CO_2$ concentration). We use two hidden-layer neural networks with hyperbolic tangent activations and 50 units each. We augment the input of the neural networks with an additional sine and cosine transformation of the features i.e., we use the feature vectors $(t_i, \sin(2\pi t_i/T), \cos(2\pi t_i/T))$ where $t_i$ is the time index and $T$ is the period used in Rasmussen and Williams [13]. For FSP-LAPLACE, we use 100 context points placed on a regular grid and limit Lanczos to run at most 500 iterations. For FVI, we draw uniformly 100 context points at each update. For Laplace, we use the full generalized Gauss-Newton matrix and find the prior scale by maximizing the marginal likelihood [11].

**Ocean current modeling.** We consider the GulfDrifters dataset [31] and we follow the setup by Shalashilin [32]. We use as training data 20 simulated velocity measurements (red arrows in Figure 2), and consider as testing data 544 average velocity measurements (blue arrows in Figure 2) computed over a regular grid on the $[-90.8, -83.8] \times [24.0, 27.5]$ longitude-latitude interval. We standardize both features and regression targets. We incorporate physical properties of ocean currents into the models by applying the Helmholtz decomposition to the GP prior's covariance function [33] as well as to the neural network $f$ using the following parameterization

$$f(\cdot, \boldsymbol{w}) = \operatorname{grad} \Phi(\cdot, \boldsymbol{w}_1) + \operatorname{rot} \Psi(\cdot, \boldsymbol{w}_2) \tag{B.2}$$

where $\boldsymbol{w} = \{\boldsymbol{w}_1, \boldsymbol{w}_2\}$ and $\Phi(\cdot, \boldsymbol{w}_1)$ and $\Psi(\cdot, \boldsymbol{w}_2)$ are two hidden-layer fully-connected neural networks with hyperbolic tangent activation functions and 100 hidden units per layer. We use 96 context points placed on a regular grid for the FSP-LAPLACE and limit Lanczos to run at most 500 iterations. For FVI, we use the same 96 context points. For the linearized Laplace, we use the full generalized Gauss-Newton (GGN) and fix the prior scale to $\sigma_p = 1$.

**Image classification.** We consider the MNIST [34] and FashionMNIST [35] image classification datasets. We standardizing the images, fit the models on a random partition of $90\%$ of the provided train splits, keeping the remaining $10\%$ as validation data, and evaluate the models on the test data. We report mean and standard-errors across 5 such random partitions of the train data with different random seeds. We compare our model to FVI, Laplace, MAP, and Sparse GP baselines, as the scale of the datasets forbids exact GP inference. The expected log-likelihood and expected calibration error are estimated by Monte Carlo integration with 10 posterior samples. All neural networks have the same convolutional neural network architecture with three convolutional layers ($3 \times 3$ kernels and output channels 16, 32 and 64) interleaved with a max-pooling layer, before two fully connected layers (with output size 128 and 10). For FSP-LAPLACE, we sample context points from the Kuzushiji-MNIST (KMNIST) dataset [62] of $28 \times 28$ gray-scale images during training following Rudner et al. [26] and use $25'000$ points from the Halton low discrepancy sequence [63] to compute the covariance. We also consider sample context points uniformly over the range $[p_{min}^{h,w,c}, p_{max}^{h,w,c}]_{h=1,w=1,c=1}^{H,W,C}$, where $H$, $W$ and $C$ are respectively the height, width and number of channels of the images, and $p_{min}^{h,w,c} = v_{min}^{h,w,c} - 0.5 \times \Delta^{h,w,c}$ and $p_{max}^{h,w,c} = v_{max}^{h,w,c} + 0.5 \times \Delta^{h,w,c}$ where $\Delta^{h,w,c} = v_{max}^{h,w,c} - v_{min}^{h,w,c}$ is the difference between the minimal ($v_{min}^{h,w,c}$) and maximal ($v_{max}^{h,w,c}$) values of the data set at pixel index $(h, w, c)$. For FVI, we use the same context point distributions as FSP-LAPLACE. For the Laplace, we use the K-FAC approximation of the generalized Gauss-Newton matrix. We use a Categorical likelihood, the Adam optimizer [64] with a batch size of 100 and stop training early when the validation loss stops decreasing. We optimize hyper-parameters using the Bayesian optimization tool provided by Weights and Biases [65] and select the parameters which maximize the average validation expected log-likelihood across 1 random partitioning of the provided training split into training and validation data. We find covariance function parameters by maximizing the log-marginal likelihood from batches [66] using the reparameterization of classifications labels into regression targets from Milios et al. [61]. We optimize over kernel, prior scale, learning-rate, $\alpha_\epsilon$ (introduced by Milios et al. [61]) and activation function and select covariance functions among the RBF, Matern-1/2, Matern-3/2, Matern-5/2 and Rational Quadratic.

**Out-of-distribution detection with image data.** We consider out-of-distribution detection with image data and a Categorical likelihood following the setup by Osawa et al. [36]. We aim to partition in-distribution (ID) data from out-of-distribution (OOD) based on the mean of entropy of the predictive distribution with respect to 10 posterior samples using a single threshold found by fitting a decision stump. We evaluate a model fit on MNIST using its test data set as in-distribution data (ID) and the test data set of FashionMNIST as out-of-distribution (OOD), and vice-versa for a model fit on FashionMNIST. We use the same models and hyper-parameters as for image classification and report mean and standard-error of our scores across the same 5 random partitions of the data.

**Bayesian optimization.** We consider Bayesian optimization (BO) problems derived from Li et al. [7]. More specifically, we use the same setup but change the dimension of the feature space of the tasks. We report mean and standard error of 5 repetitions of the tasks across different random seeds. We use two hidden layer neural networks with hyperbolic tangent activations and 50 hidden units each. FSP-LAPLACE uses a Matern-5/2 covariance function with constant zero mean function whose parameters are found by maximizing the marginal likelihood [13]. We use 400 context points during

training and $10'000$ to compute the posterior covariance sampled using latin hypercube sampling [67]. We use the same prior for FVI and the same number of context points during training. The Gaussian process uses a zero mean function and a Matern-5/2 covariance function following Li et al. [7]. For the Laplace approximation, we find the prior scale by maximizing the marginal likelihood [11]. Unlike FSP-LAPLACE which uses low-rank factors to parameterize the posterior covariance, we found that repeatedly computing the covariance and predictive posterior of the linearized Laplace with the full and K-FAC generalized Gauss-Newton (GGN) matrices was often prohibitively slow in the BO setup. We therefore use the K-FAC approximation to the GGN where possible (Branin and PDE) and the diagonal approximation otherwise (Ackley, Hartmann, Polynomial and BNN). We implemented this experiment using the BO Torch library [68].

**Regression with UCI datasets.** We consider tabular regression datasets from the UCI repository [69]. Specifically, we perform leave-one-out 5-fold cross validation, considering $10\%$ of the training folds as validation data, and we report the mean and standard-error of the average expected log-likelihood on the test fold. We report the mean rank of the methods across all datasets by assigning rank 1 to the best scoring method as well as any method who's standard error overlaps with the highest score's error bars, and recursively apply this procedure to the methods not having yet been assigned a rank. We estimate the expected log-likelihood using 10 posterior samples. We encoding categorical features as one-hot vectors and standardizing the features and labels. We consider two hidden-layer neural networks with 50 hidden units each and hyperbolic tangent activations. All models have a homoskedastic noise model with a learned scale parameter. FSP-LAPLACE uses context points drawn uniformly over the range $[p^i_{min}, p^i_{max}]^d_{i=1}$, where $d$ is the dimension of the feature space, and $p^i_{min} = v^i_{min} - 0.5 \times \Delta^i$ and $p^i_{max} = v^i_{max} + 0.5 \times \Delta^i$ where $\Delta^i = v^i_{max} - v^i_{min}$ is the difference between the minimal ($v^i_{min}$) and maximal ($v^i_{max}$) values of the data set at feature index $i$. For the Laplace, we use the full generalized Gauss-Newton matrix. FVI uses the same context points as FSP-LAPLACE. Neural networks are fit using the Adam optimizer [64] and we stop training early when the validation loss stops decreasing. Hyper-parameters are found just as for the image classification experiment.

**Out-of-distribution detection with regression data.** We consider out-of-distribution (OOD) detection with tabular regression data from the UCI datasets [69] following the setup from Malinin et al. [70]. We aim to separate test data (in-distribution) from a subset of the song dataset [71] (out-of-distribution) with the same number of features based on the variance of the predictive posterior estimated from 10 posterior samples using a single threshold obtained via a decision stump. We process the data just like in the regression experiments, use the same model hyper-parameters and report mean and standard-error of the scores across the same 5 random partitions of the data.

### B.2.1 Software

We use the JAX [72] and DM-Haiku [73] Python libraries to implement neural networks. The generalized Gauss-Newton matrices used in the Laplace approximations are computed using the KFAC-JAX library [74]. We implemented the GPs and sparse GPs using the GPyTorch library [68]. We conducted experiments the Bayesian optimization experiments using the BOTorch library [68].

### B.2.2 Hardware

All models were fit using a single NVIDIA RTX 2080Ti GPU with 11GB of memory.

## C Additional experimental results

### C.1 Additional qualitative results

**Regression.** We show additional results for the synthetic regression task described in Section 4. We find that FSP-LAPLACE successfully adapts to the beliefs specified by different Gaussian process priors in terms of periodicity (Figure C.2), smoothness (Figure 1) and length scale (Figure C.3) without modifying the neural network's architecture or adding features. We also find that our method effectively regularizes the model when the data is very noisy (Figure C.4).
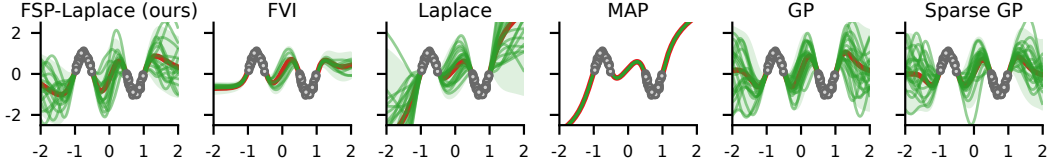
Figure C.1: Just like the Gaussian process (GP) and sparse GP, FSP-LAPLACE captures the smoothness behavior specified by the RBF covariance function of the Gaussian process prior.
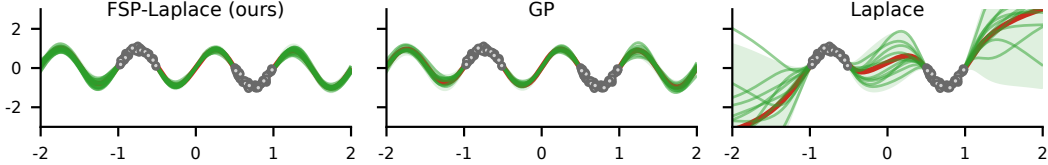


Figure C.2: Unlike the linearized Laplace, FSP-LAPLACE allows to incorporate periodicity within the support of the data using a periodic prior covariance function and without additional periodic features.
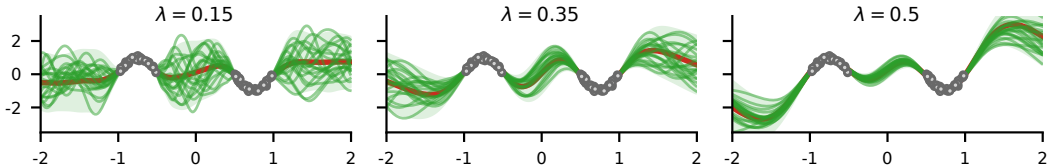


Figure C.3: FSP-LAPLACE adapts to the length scale provided by the RBF covariance function of the Gaussian process prior.
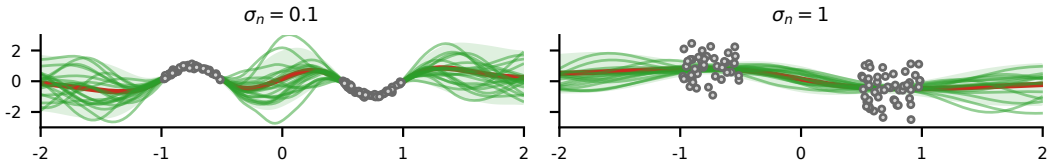


Figure C.4: FSP-LAPLACE is effectively regularized under strong label noise.

**Classification.** We show additional results for the two-moons classification task described in Section 4. Similar to the GP and sparse GP baselines, we find that our method captures the behavior of the prior, showing a smooth decision boundary when equipped with a RBF covariance function (Figure C.6) and a rough decision boundary when equipped with a Matern-1/2 covariance function (Figure C.5). FSP-LAPLACE also reverts to the zero-valued mean outside of the data support.

**Effect of context points.** We provide additional details on the role of the context points in FSP-LAPLACE. The goal of the context points is to regularize the neural network on a finite set of input locations which includes any point where we would like to evaluate the model. During MAP estimation (see Algorithm 1), context points are resampled at each update step to amortize the coverage of the feature space. During the posterior covariance computation (see Algorithm 2), context points are fixed and define where we regularize the model. Context points bare similarity with inducing points in variational GPs [27] in this later step as both define where the model is regularized.

We show additional results demonstrating the behavior of our model in the low context point regime. Figure C.7 shows the effect of the number of context points on a 1-dimensional regression task with GP priors equipped with a RBF and a Matern 1/2 kernel. Figure C.8 shows the same experiment but on a 2-dimensional classification task. The $M$ context points are randomly sampled uniformly during training, and we use the Halton low discrepancy sequence as context points to compute the covariance. Even with a very small number of context points ($M = 3$ and $M = 5$), our model still produces useful uncertainty estimates even if it cannot accurately capture the beliefs specified by the
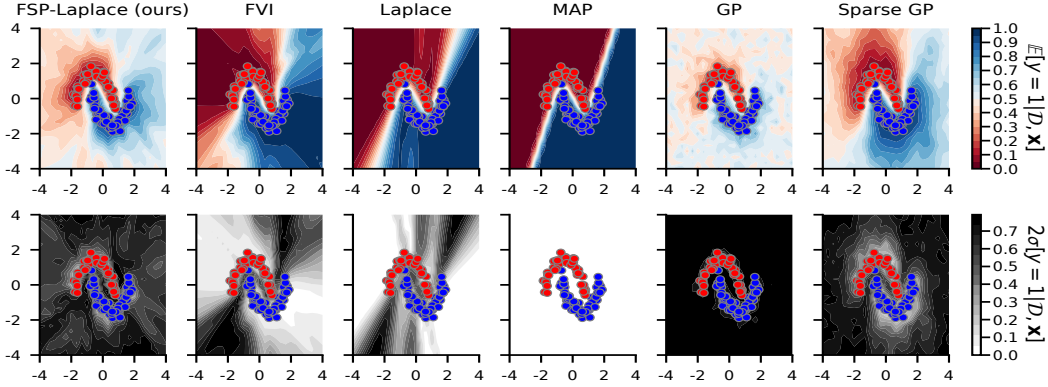
Figure C.5: FSP-LAPLACE with a Matern-1/2 covariance function against baselines in the two-moons classification task. Similar to the Gaussian process (GP) and sparse GP, our method shows a rough decision boundary.
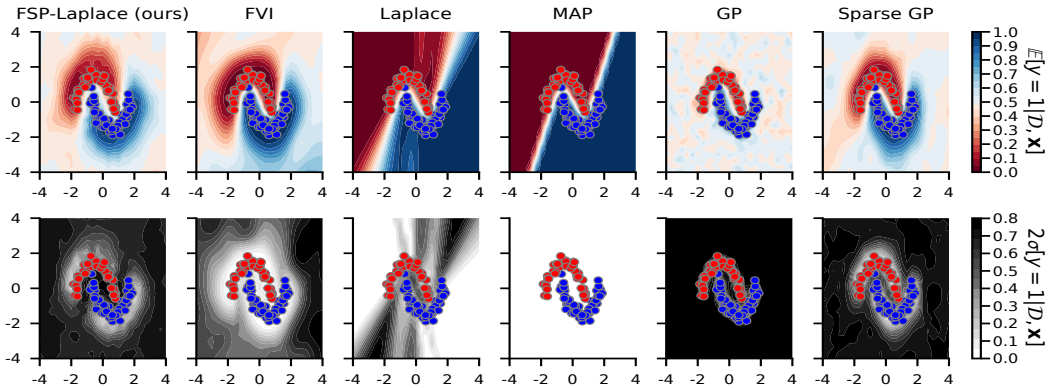


Figure C.6: FSP-LAPLACE with a RBF covariance function against baselines in the two-moons classification task. Similar to the Gaussian process (GP) and sparse GP, our method shows a smooth decision boundary.

prior. We also note that our method requires more context points to capture the beliefs of rougher priors than smooth priors (see Figure C.7).

## C.2 Additional quantitative results

**Mauna Loa.** We here show the figure associated with the Mauna Loa dataset experiment in Section 4.1. Figure C.9 shows the predictions of FSP-LAPLACE and the baselines on the Mauna Loa dataset. We find that incorporating prior beliefs both via an informative prior and periodic features in FSP-LAPLACE results in an improved fit over FVI, Laplace and GP baselines.
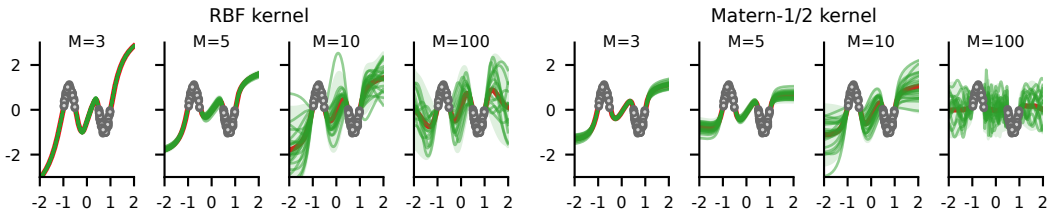


Figure C.7: FSP-LAPLACE with a smooth RBF covariance function and a rough Matern-1/2 with varying amounts of context points $M$. Given a very small number of context points, our method still produces useful uncertainty estimates.
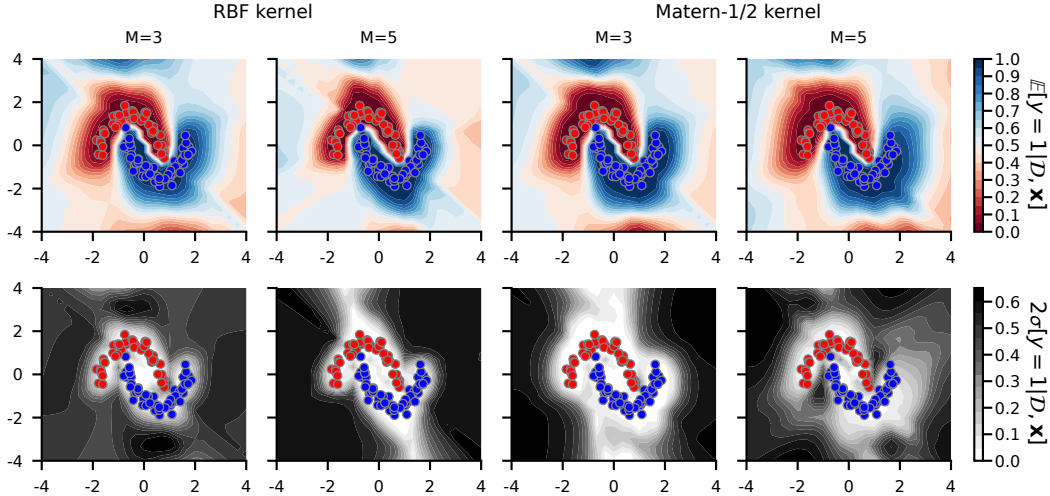
Figure C.8: FSP-LAPLACE with a smooth RBF covariance function and a rough Matern-1/2 with varying amounts of context points $M$. Given a very small number of context points, our method still produces useful uncertainty estimates and reverts to the prior's zero valued mean.
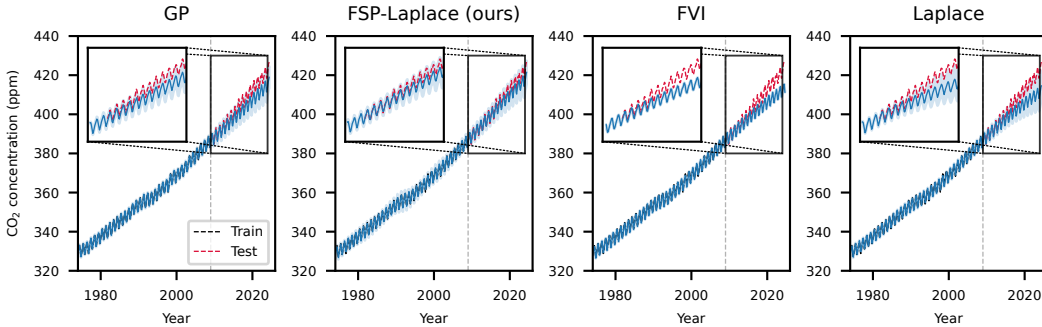


Figure C.9: Regression on the Mauna Loa dataset. Incorporating prior knowledge via a kernel tailored specifically to the dataset, our method (FSP-LAPLACE) results in a strong decrease in mean square error over baselines.

**Out-of-distribution detection with image data.** We present additional results for the out-of-distribution detection experiment presented in Section 4.1. Figure C.10 shows the distribution of the predictive entropy of in-distribution (ID) and out-of-distribution (OOD) data under the sparse GP, FSP-LAPLACE, FVI and Laplace models. For both MNIST and FashionMNIST, we find that the predictive entropy of ID data under FSP-LAPLACE is tightly peaked around $0$ nats and that the predictive entropy of OOD data strongly concentrates around its maximum $\ln 10 \approx 2.3$ nats.

**Rotated MNIST and FashionMNIST.** We provide an additional experiment studying the behavior of FSP-LAPLACE under out-of-distribution data. We consider the setup by Rudner et al. [25], Sensoy et al. [75] and track the predictive entropy of models trained on MNIST and FashionMNIST for increasing angles of rotation of the test images. We expect the predictive entropy of a well-calibrated neural network to grow as the inputs become increasingly dissimilar to the training data with higher angles of rotation. Similar to FVI, sparse GP and the linearized Laplace baselines, we find that FSP-LAPLACE yields low predictive entropy for small rotation angles and that the predictive entropy increases with the angle on both MNIST and FashionMNIST which is what we expect from a well calibrated Bayesian model.

**Regression with UCI datasets.** We further evaluate our method on regression datasets from the UCI repository [69] and compare FSP-LAPLACE to FVI, Laplace, MAP, GP, and Sparse GP baselines. We perform leave-one-out 5-fold cross-validation, keeping $20\%$ of the remaining train folds as validation data. We report the mean and standard-error of the expected log-likelihood with respect to samples
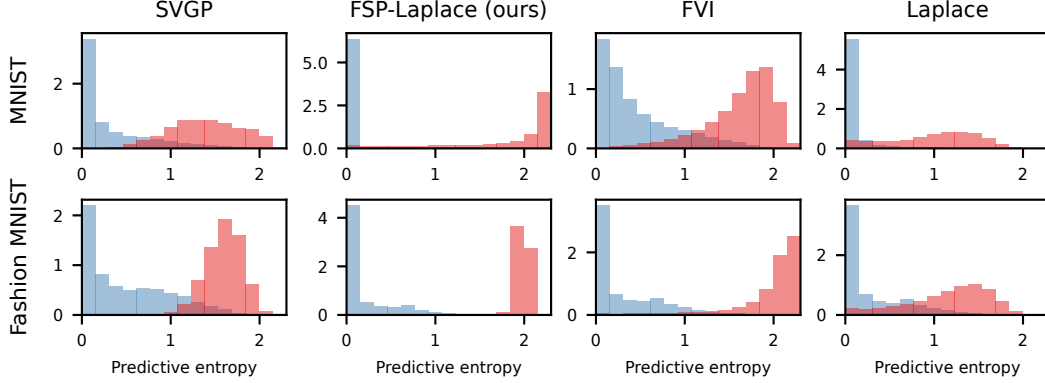
22

Figure C.10: Distribution of in-distribution (ID) and out-of-distribution (OOD) samples for the MNIST and Fashion MNIST image datasets. The predictive entropy produced by FSP-LAPLACE nearly perfectly partition ID and OOD data. This is reflected in OOD accuracy in Table 2.
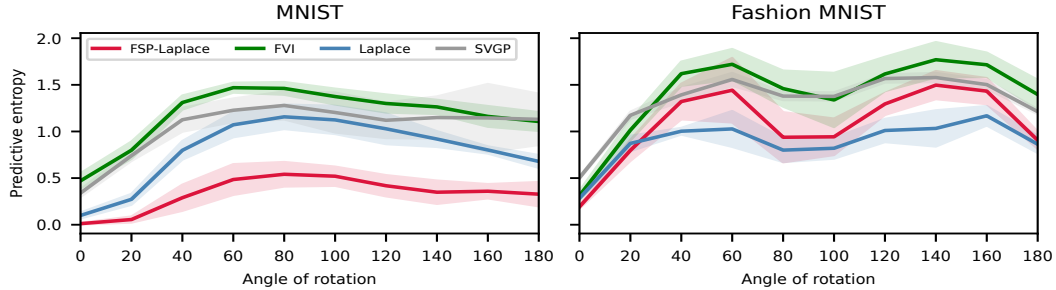


Figure C.11: Input distribution shift experiment. We find that FSP-LAPLACE shows greater predictive entropy as the input becomes increasingly dissimilar to the training data.

from the posterior across the 5-folds. Additional details can be found in Appendix B.2. Results are presented in Table C.1. We find that our method performs well compared to baselines, matching or improving over the mean rank of all Bayesian methods (FVI, Laplace, GP, and Sparse GP) but a slightly lower mean rank than the MAP baseline (1.636 vs. 1.363). In particular, our method is noticeably more accurate than the Laplace.

Table C.1: Test log-likelihood and out-of-distribution accuracy of evaluated methods on regression datasets from the UCI repository. We find that our method matches or improves over the mean rank of Bayesian baselines in terms of expected log-likelihood, and obtains out-of-distribution detection accuracies similar to the best performing BNN.

| DATASET | EXPECTED LOG-LIKELIHOOD (↑) | | | | | | OUT-OF-DISTRIBUTION DETECTION ACCURACY (↑) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | FSP-LAPLACE (OURS) | FVI | LAPLACE | MAP | GP | SPARSE GP | FSP-LAPLACE (OURS) | FVI | LAPLACE | GP | SPARSE GP |
| BOSTON | **-0.641 ± 0.084** | **-0.512 ± 0.105** | -0.733 ± 0.019 | -0.906 ± 0.274 | -1.136 ± 0.035 | -0.762 ± 0.077 | 0.923 ± 0.015 | 0.754 ± 0.019 | 0.913 ± 0.008 | 0.948 ± 0.004 | **0.969 ± 0.003** |
| CONCRETE | -0.352 ± 0.052 | -0.378 ± 0.076 | -0.538 ± 0.026 | **-0.199 ± 0.056** | -0.766 ± 0.119 | -0.609 ± 0.050 | **0.914 ± 0.007** | 0.585 ± 0.011 | 0.859 ± 0.002 | 0.878 ± 0.005 | 0.854 ± 0.007 |
| ENERGY | 1.403 ± 0.033 | 1.448 ± 0.059 | 1.423 ± 0.062 | **1.754 ± 0.058** | 1.370 ± 0.060 | 1.518 ± 0.032 | **1.000 ± 0.000** | 0.928 ± 0.007 | **1.000 ± 0.000** | **1.000 ± 0.000** | **1.000 ± 0.000** |
| KIN8NM | -0.134 ± 0.010 | **-0.135 ± 0.021** | -0.199 ± 0.009 | **-0.105 ± 0.011** | - | -0.206 ± 0.003 | 0.626 ± 0.006 | 0.593 ± 0.006 | 0.609 ± 0.015 | - | **0.645 ± 0.004** |
| NAVAL | **3.482 ± 0.014** | 2.943 ± 0.062 | 3.213 ± 0.028 | **3.460 ± 0.048** | - | **3.474 ± 0.019** | **1.000 ± 0.000** | **1.000 ± 0.000** | **1.000 ± 0.000** | - | **1.000 ± 0.000** |
| POWER | 0.045 ± 0.014 | **0.071 ± 0.018** | 0.016 ± 0.011 | **0.047 ± 0.013** | - | 0.026 ± 0.009 | **0.818 ± 0.003** | 0.686 ± 0.014 | **0.820 ± 0.003** | - | 0.777 ± 0.004 |
| PROTEIN | -0.991 ± 0.005 | **-0.988 ± 0.002** | -1.043 ± 0.006 | -1.003 ± 0.003 | - | -1.035 ± 0.002 | 0.917 ± 0.004 | 0.857 ± 0.015 | **0.992 ± 0.001** | - | 0.967 ± 0.001 |
| WINE | -1.189 ± 0.019 | **-1.193 ± 0.015** | -1.403 ± 0.014 | **-1.174 ± 0.022** | -1.313 ± 0.032 | -1.194 ± 0.020 | 0.591 ± 0.024 | 0.541 ± 0.012 | 0.763 ± 0.012 | **0.812 ± 0.008** | 0.680 ± 0.014 |
| YACHT | 0.183 ± 0.438 | 0.655 ± 0.293 | 0.136 ± 0.762 | **1.644 ± 0.310** | 1.336 ± 0.060 | 1.532 ± 0.139 | **0.973 ± 0.008** | 0.594 ± 0.010 | 0.851 ± 0.029 | 0.842 ± 0.023 | 0.651 ± 0.011 |
| WAVE | 6.112 ± 0.049 | 6.552 ± 0.155 | 6.460 ± 0.025 | **7.146 ± 0.195** | - | 4.909 ± 0.001 | **0.912 ± 0.076** | **0.861 ± 0.014** | 0.810 ± 0.023 | - | 0.513 ± 0.001 |
| DENMARK | **-0.364 ± 0.008** | -0.462 ± 0.006 | -0.810 ± 0.010 | -0.693 ± 0.014 | - | -0.768 ± 0.001 | 0.677 ± 0.004 | 0.634 ± 0.008 | **0.714 ± 0.006** | - | 0.626 ± 0.002 |
| MEAN RANK | 1.636 | 1.636 | 2.727 | 1.363 | 2.600 | 2.455 | 1.818 | 3.091 | 1.727 | 1.6 | 2.182 |

**Out-of-distribution detection on tabular data.** We also investigate whether the epistemic uncertainty of FSP-LAPLACE is predictive of out-of-distribution data in the regression setting by evaluating it on out-of-distribution detection following Malinin et al. [70]. We report the accuracy of a single threshold to classify OOD from in-distribution (ID) data based on the predictive uncertainty. More details are presented in Appendix B.2. In the context of tabular data, we find that FSP-LAPLACE performs second best among BNNs and is almost as accurate as the Laplace approximation (see Table C.1) which is first. We note that FSP-LAPLACE systematically outperforms FVI in terms of out-of-distribution detection and obtains a higher mean rank (1.818 vs. 3.091).

$\|P_0 \Lambda P_0^\top\|_F$ **is negligible.** We provide evidence that the term $\|P_0 \Lambda P_0^\top\|_F$ in Section 3.2 is negligible compared to $\|\Lambda\|_F$ in four different configurations. We consider the synthetic regression and classification setups described in Appendix B.1.

Table C.2: $\|P_0 \Lambda P_0^\top\|_F / \|\Lambda\|_F$ for different combinations of priors and tasks.

| KERNEL | REGRESSION | CLASSIFICATION |
|---|---|---|
| RBF | $1.026 \times 10^{-7}$ | $3.548 \times 10^{-4}$ |
| MATERN-1/2 | $4.305 \times 10^{-6}$ | $1.299 \times 10^{-2}$ |

# NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our experimental results support the fact that the our methods captures beliefs specified by the GP prior.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in sections Sections 3, 4 and 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We rigorously detail assumptions in Appendix A.1 and proofs can be found in Appendix A.2 and the references.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We do so in Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code will be published at https://github.com/tristancinquin/fsplaplace.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: See Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report standard error along the mean scores across cross validation folds.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: See Appendix Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: To the best of our knowledge, we have fully conformed with the code of ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: We believe that there is no direct societal impact of the work perform that needs discussion.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The methods developed in the paper have no risk for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All external assets (code and data) used in this publication are publicly available, cited in Section 4 and Appendix C.1, and their respective licenses are, to the best of our knowledge, properly respected.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not conduct any research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not conduct any research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.