Functional Variational Inference based on Stochastic Process Generators

Chao Ma
University of Cambridge
Cambridge, UK
cm905@cam.ac.uk

José Miguel Hernández-Lobato
University of Cambridge
Cambridge, UK
jmh233@cam.ac.uk

Abstract

Bayesian inference in the space of functions has been an important topic for Bayesian modeling in the past. In this paper, we propose a new solution to this problem called Functional Variational Inference (FVI). In FVI, we minimize a divergence in function space between the variational distribution and the posterior process. This is done by using as functional variational family a new class of flexible distributions called Stochastic Process Generators (SPGs), which are cleverly designed so that the functional ELBO can be estimated efficiently using analytic solutions and mini-batch sampling. FVI can be applied to stochastic process priors when random function samples from those priors are available. Our experiments show that FVI consistently outperforms weight-space and function space VI methods on several tasks, which validates the effectiveness of our approach.

1 Introduction

As an important approach to Bayesian deep learning, Bayesian neural networks (BNNs) have been proposed and studied for decades [29, 34, 19]. Despite some successful applications in specific cases [55, 44, 23], BNNs still suffer from several shortcomings. As over-parameterized models, BNNs often have multiple posterior modes in its weight space, that generate identical predictions [30]. Therefore, Bayesian inference in weight space can be very difficult [50]. Furthermore, due to the complexity of interactions between weights in the network forward pass, it is unclear the effect that a given prior distribution over the weights will have in the resulting distribution over functions.

To address these issues, there has been a recent resurgence of interest in applying the perspective of Bayesian non-parametrics to neural nets. That is, BNNs are treated as probability measures over functions, i.e., as stochastic processes, with Bayesian inference being performed now in function space. Examples include functional BNNs (f-BNNs) [47], and variational implicit processes (VIPs) [28]. Despite their empirical advantage over weight space VI, they still suffer from a number of issues. i), f-BNNs optimize a functional KL divergence between BNNs and GPs, which is not always well-defined [5]. Also, f-BNNs rely on (spectral) stein gradient estimators, which are less efficient for high dimensional distributions [56, 14]. While VIPs do not have this problem, they use a wake-sleep procedure that does not correspond to a single unified objective. ii), F-BNNs rely on a mean-field approximation, which often lacks predictive in-between uncertainty on test data. This issue was observed both with single layer BNNs [10], and deeper BNNs (able to represent in-between uncertainty, but more over-confident than HMC empirically for certain settings [11, 9]). On the other hand, VIPs use as variational family a Gaussian process, which is not able to capture non-Gaussian processes such as structured implicit priors [47].

Therefore, it is an open challenge how to improve and justify (variational) inference in the space of functions using priors given by stochastic processes. In this paper, we investigate this old but important problem and propose a new solution. Our contributions are as follows:
We propose a new objective function for variational inference in function space, as an alternative to functional KL divergence between stochastic processes [47]. We show that this new objective is a valid divergence, and can avoid some of the problems that the functional KL divergence has.

We propose a new class of flexible variational distributions in function space, called stochastic process generators (SPGs). SPGs are non-Gaussian generalizations of the VIP family [28], and can help avoid the fundamental limitation of mean field Gaussians [10] used in BNNs and functional BNNs. A theorem regarding the expressiveness of SPGs is proved (Proposition 4). Based on SPGs, our proposed functional divergence between stochastic processes can be estimated efficiently using mini-batch sampling (Proposition 5 and 7), which achieves a significant speed-up against the gradient estimator approach in [47].

We compare our methods against existing weight-space and function-space inference methods in several tasks. Our method consistently outperforms the baselines, and is much faster than f-BNN, which validates the effectiveness of our approach.

## 2 Backgrounds

**Variational BNNs** Let \( y = g(x, w) \) be a neural network, where \( x \) is the input, \( y \) is the output, and \( w \) denotes the weights. To build a Bayesian neural network (BNN), we place a prior \( p(w) \) over \( w \), and add an observational noise \( \epsilon \sim N(\epsilon; 0, \sigma^2) \) to the output. Then, a BNN is given by

\[
\log p(y|x) = \log \int_w N(y; g(x, w), \sigma^2)p(w)dw. \tag{1}
\]

Given a dataset \( D = \{x_i, y_i\}_{i=1}^N \), the goal of Bayesian inference is to compute the posterior \( p(w|D) \propto p(D|w)p(w) \). Since this is intractable, we often resort to approximate inference methods such as variational inference (VI), e.g. Bayes by Backprop (BBB) [4], variational dropout [12], or other extensions [20, 26, 41]. Since the posterior \( p(w|D) \) is usually highly multi-modal due to the non-identifiability of neural networks, VI often leads to unsatisfactory results, and suffers from over-parameterization and other pathologies [10].

**Variational Implicit Processes (VIPs)** An alternative way of looking at BNNs is to treat them as a prior distribution over function space, i.e., a stochastic process. Indeed, \( p(y|x) \) in Equation 1 generates a random function \( g(\cdot, w) \), depending on what \( w \) is actually sampled from the prior. This idea is further explored by the variational implicit process (VIPs). Given a prior \( p(f) \) on random functions, VIP approximates the posterior in function space \( p(f|D) \) by using a Bayesian linear regression model \( q_{VIP}(f|D) = \sum_s a_s \phi_s(a) \sim N(\mu, \Sigma) \), where the basis functions \( \{\phi_s\}_{s=1}^S \) are random samples drawn directly from \( p(f) \). Although VIP has shown improved performance over parameter-space VI methods, it has a few limitations. For example, \( q_{VIP}(f|D) \) resembles a GP approximation to \( p(f|D) \), whereas the true posterior in function space might be arbitrarily complex. Also, the wake-sleep training procedure of VIP does not optimize any coherent objective function. Lastly, VIP requires the prior \( p(f) \) to be reparameterizable. i.e., can be represented in the form of \( f(x) = g(x, z), z \sim p(z) \). This may limit its application to non-reparameterizable priors.
3 Problem setting and the functional KL divergence

We consider the problem of Bayesian inference in function space. Let \( p(f) \) be a stochastic process defined on the probability space \((\Omega, \mathcal{B})\). Note that we use \( f \) in its scalar form to denote a scalar function \( f(\cdot) : \mathcal{T} \rightarrow \mathbb{R} \). Here \( \mathcal{T} \) is the index set of \( p(f) \) (assumed to be a compact subset of \( \mathbb{R}^d \)). For example, \( p(f) \) could be a Gaussian process \( p(f) = \mathcal{GP}(m(\cdot), k(\cdot, \cdot)) \), a Bayesian neural network, or any other suitable stochastic process. We use \( p(f) \) to model the uncertainty in function space. Then, a likelihood function \( p_n(y|f(\cdot)) \) is defined on top of \( f \) to generate observable data \( y \).

Given observed data \( D = \{x_i, y_i\}_{i=1}^N \), our goal is to infer the posterior process \( p(f|D) \) conditioned on the observations \( D \). If \( p(f) \) is a GP, then \( p(f|D) \) can be computed analytically. However, in most cases this is intractable. Therefore, following [32, 47], we define another stochastic process \( q(f) \) on \((\Omega, \mathcal{B})\) as our variational family to approximate \( p(f|D) \). \( q(f) \) can be optimized by maximizing the evidence lower bound (ELBO) in function space:

\[
\mathcal{L}_q^{\text{functional}} := \mathbb{E}_{q(f)}[\log p_x(D|f)] - D_{KL}[q(f)||p(f)].
\]  

(2)

Note that \( D_{KL}[q(f)||p(f)] \) is the functional KL-divergence between stochastic processes. As shown by [47], \( D_{KL}[q(f)||p(f)] \) can be written as:

\[
D_{KL}[q(f)||p(f)] = \sup_{f_n, X_n} D_{KL}[q(f_{X_n})||p(f_{X_n})],
\]  

(3)

where \( X_n \) denote a set of \( n \) measure points \( \{x_k\}_{1 \leq k \leq n} \) in the domain/index set of \( f(\cdot) \), which can be treated as an element of the product space \( \mathcal{T}^n \). Moreover, \( f_{X_n} \) denotes the vector of function values evaluated on \( X_n \), and \( D_{KL}[q(f_{X_n})||p(f_{X_n})] \) is the KL-divergence over random vectors typically used by the machine learning community. In other words, the KL-divergence between stochastic processes is the supreme of the relative entropies obtained on all possible measure points in \( \mathcal{T}^Z \).

4 Functional Variational Inference using Stochastic Process Generators

4.1 The grid-functional KL divergence

As pointed out by [3], the functional KL divergence defined in Eq 2 is not always well-defined. For example, the functional KL divergence between two BNNs with different network architectures can be infinite. To address this issue, we propose to optimize a new divergence measure called the grid-functional KL divergence. Figure 2 illustrates the idea of grid functional KL divergence, which we now proceed to describe. Instead of taking the supremum as in the original functional KL divergence, we take an expectation over both \( n \) and \( X_n \). We specify a probability distribution \( \{p_n\}_{1 \leq n < \infty} \) over \( n \) assigning lower probability values \( p_n \) to larger \( n \). This way, we can trim down the contributions from the KL terms \( D_{KL}[q(f_{X_n})||p(f_{X_n}|D)] \) that correspond to large \( n \), and hopefully arrive at a finite expectation value. Following this idea, we give the formal definition of grid-functional KL divergence as

\[
D_{grid}[q(f)||p(f|D)] := \mathbb{E}_{p_0, X_n \sim \gamma} D_{KL}[q(f_{X_n})||p(f_{X_n}|D)],
\]  

(4)

where \( X_n \) is a set of \( n \) measurement points \( \{x_k\}_{1 \leq k \leq n} \) sampled from \( \mathcal{T} \), according to some sampling distribution \( c \). Note that \( c \) is defined on the product space \( \mathcal{T}^Z \) and the number of sampled measure points \( n \) is also random. One may recognize that [47] proposed a similar objective as an approximation to [3] in which the number of measure points \( n \) is a fixed constant instead of a random variable. However, as we show below, it is in fact critical that \( n \) is not fixed.

**Proposition 1.** Suppose \( c \) has full support on \( \mathcal{T}^Z \). Then, \( D_{grid}[q(f)||p(f|D)] \) satisfies the following conditions: i) \( D_{grid}[q(f)||p(f|D)] \geq 0 \); ii) \( D_{grid}[q(f)||p(f|D)] = 0 \) if and only if \( q(f) = p(f|D) \).
In other words, if \( c \) has full support on \( \mathcal{T}^{z^+} \) (we will give an example of such \( c \) later), then 
\[
D_{\text{grid}}[q(f) || p(f|D)]
\]
is a valid divergence in function space. Therefore, we can use \( D_{\text{grid}} \) as an alternative objective for function space inference. We can show that, for certain scenarios, \( D_{\text{grid}} \) can avoid some of the issues the original functional KL divergence has (see Appendix A.2 for details):

**Proposition 2.** Let \( p(f) \) and \( q(f) \) be two distributions for random functions with different parametric forms. Assume that \( p(f) \) is parameterized by the following sampling processes:

\[
f = g + \epsilon, \quad g(x) \sim p(g|x; \Theta), \quad \Theta \sim p(\Theta), \quad \epsilon \sim \mathcal{N}(0, \sigma^2)
\]

and \( q(f) \) is parameterized by:

\[
f = g + \epsilon, \quad g(x) \sim q(g|x; \Gamma), \quad \Gamma \sim q(\Gamma), \quad \epsilon \sim \mathcal{N}(0, \sigma^2).
\]

Here, \( x \in \mathcal{T} \subset \mathbb{R}^d \) is the input variable of the \( f \), \( g \) is the random latent function, and \( \Theta \in \mathbb{R}^l \), \( \Gamma \in \mathbb{R}^j \) are the parameters of each random function, respectively. Suppose \( p(g|x; \Theta) \), \( q(g|x; \Gamma) \), \( p(\Theta) \) and \( q(\Gamma) \) all have compact supports w.r.t. \( g, \Theta, \Gamma \), respectively. Then, there exist a sampling distribution \( c \) such that: 1) \( c \) has full support on \( \mathcal{T}^{z^+} \), and 2) \( D_{\text{grid}}[q(f) || p(f)] \) is finite.

This result shows that the grid-functional KL divergence allows us to perform VI between \( p \) and \( q \) even if they have different parametric forms. In Appendix A.2, we further show that the compactness assumptions can be relaxed under certain assumptions. Moreover, in Appendix A.3 Corollary 1, we have shown that if one of the distributions is replaced by a Gaussian process (for certain kernel assumptions), then under some additional assumptions, the grid-functional KL is still finite. In those cases, the original functional KL is no longer finite. This validates our choice of grid-functional KL divergence. To use \( D_{\text{grid}}[q(f) || p(f)] \) for VI, we further derive a new ELBO based on \( D_{\text{grid}} \) (Appendix A.1):

**Proposition 3.** Let \( n, X_n \sim c \) be a set of random measure points such that \( X_D \subset X_n \). Define

\[
\mathcal{L}^\text{grid}_q := \log p(D) - D_{\text{grid}}[q(f) || p(f|D)],
\]

Then we have

\[
\mathcal{L}^\text{grid}_q = \mathbb{E}_q(f) [\log p(D|f)] - D_{\text{grid}}[q(f) || p(f)]
\]

and \( \log p(D) \geq \mathcal{L}^\text{grid}_q \geq \mathcal{L}^\text{functional}_q \).

Proposition 3 shows that \( \mathcal{L}^\text{grid}_q \) is a valid variational objective function: it is a lower bound for \( \log p(D) \) and also upper-bounds \( \mathcal{L}^\text{functional}_q \). For the rest of the paper, we will discuss how to perform functional VI based on \( \mathcal{L}^\text{grid}_q \). We will focus on how to propose a expressive variational family \( q(f) \) and how to efficiently estimate \( D_{\text{grid}}[q(f) || p(f)] \).

**Remark.** One example of \( c \) that satisfies the requirement of Propositions 1, 2 and 3 takes the following form (which will be used throughout the paper):

\[
(n - |D|) \sim \text{Geom}(p), \quad x_k \sim \mathcal{U}(\mathcal{T}), \quad \forall 1 \leq k \leq n - |D|, \quad X_n := X_D \cup \{x_k\}_{1 \leq k \leq n - |D|},
\]

where we first sample \( n \) from a geometric distribution such that \( (n - |D|) \sim \text{Geom}(p) \) with parameter \( p \) (see Appendix A.2 for more discussion). Then, \( (n - |D|) \) out of distribution (OOD) measure points are sampled independently from a uniform distribution on \( \mathcal{T} \).

### 4.2 Choosing \( q(f) \): stochastic process generators

In order to obtain good performance in FVI, it is crucial to pick an expressive variational family for \( q(f) \). Here, we propose a new class of variational distributions called stochastic process generator (SPG). SPGs can be seen as the non-Gaussian extension of the variational approximation used in VIPs. Recall that in VIPs, the variational family \( q_{\text{VIP}}(f) \) is defined by the following sampling process:

\[
f = \sum_{s} a_s \phi_s(\cdot), \quad a \sim \mathcal{N}(\mu; \Sigma),
\]

where \( \{\phi_s\}_{s=1}^{S} \) are random paths sampled from the prior process \( p(f) \). In SPGs, we remove the Gaussian assumption on \( a \). This is done by specifying a separate variational auto-encoder to represent the non-Gaussian distribution for \( a \):

\[
q(a) = \int_{h} p_\theta(a|h) q_\eta(h),
\]
where \( h \) is the latent variable of the VAE, \( p_\theta(a|h) \) is the decoder parameterized by \( \theta \), and \( q_\eta(h) \) is a distribution over the latent space. Furthermore, we remove the constraint that \( \{\phi_k\}_{k=1}^n \) need to be sampled from \( p(f) \). For example, we may assume that each \( \phi_k : \mathbb{R}^d \rightarrow \mathbb{R}^1 \) is a DNN with learnable weights \( w_s \). We can finally give the definition of our SPG variational family \( q_{\text{SPG}}(f|q_\eta(h)) \):

\[
f = \sum_s a_s \phi_s(\cdot, w_s) + \nu, \quad \nu \sim \int_h p_\theta(a|h)q_\eta(h), \quad \nu \sim \mathcal{GP}(\nu; 0, \delta(\cdot, \cdot)\sigma^2_v),
\]

where \( \nu \) is a white noise process that models additive aleatoric uncertainty. Intuitively, \( q_{\text{SPG}} \) serves as a generator for stochastic processes. \( q_{\text{SPG}}(f|\cdot) \) maps any given \( q_\eta(h) \) to a stochastic process \( q_{\text{SPG}}(f|q_\eta(h)) \), hence the name. Regarding the expressiveness of SPGs, we have the following result:

**Proposition 4** (Expressiveness of SPGs). Let \( p(f) \) be a square-integrable stochastic process on a probability space \((\mathcal{X}, \mathcal{B})\), and its index set \( \mathcal{T} \) is a compact subset of \( \mathbb{R}^d \). Suppose \( \mathcal{X} \) is a compact metric space, \( \mathcal{B} \) is the Borel set on \( \mathcal{X} \). Then, for \( \forall \epsilon > 0 \), there exists a SPG \( q_{\text{SPG}}(f) \), such that:

\[
\operatorname{MMD}(p, q_{\text{SPG}}; \mathcal{F}) < \epsilon \quad \text{for} \quad \forall x \in \mathcal{T},
\]

where \( \operatorname{MMD} \) is the maximum mean discrepancy, \( \mathcal{F} \) is the MMD function class defined to be a unit ball in a RKHS with a universal kernel \( k(\cdot, \cdot) \) as its reproducing kernel.

Next we will discuss how \( \{w_s\}, \theta \) and \( q_\eta(h) \) can be estimated and how can we use SPGs to estimate the function space KL-divergence in Equation [5].

### 4.3 Efficient estimation of grid-functional KL

In order to estimate the grid-functional KL-divergence in Equation [4] we propose to use a two-step method for a scalable approximation. In the first step, we distill \( p(f) \) by fitting a stochastic process generator \( \hat{p}_{\text{SPG}}(f) \) to \( p(f) \). In the second step, we calculate the KL-divergence between \( q_{\text{SPG}}(f) \) and \( \hat{p}_{\text{SPG}}(f) \) as the surrogate for the KL-divergence between \( q_{\text{SPG}}(f) \) and \( p(f) \).

**Distilling \( p(f) \) via a stochastic process generator.** Assume that we can draw \( M \) random functions \([f_1(\cdot), f_2(\cdot), \ldots, f_m(\cdot), \ldots, f_M(\cdot)]\) from the prior process \( p(f) \). Let \( \hat{p}_{\text{SPG}}(f) \) be an SPG given by

\[
f = \sum_s a_s \phi_s(\cdot, w_s) + \nu, \quad \nu \sim \int_h p_\theta(a|h)p_0(h), \quad \nu \sim \mathcal{GP}(\nu; 0, \delta(\cdot, \cdot)\sigma^2_v),
\]

where \( p_0(h) \) is a fixed standard normal distribution. We denote the above process by \( \hat{p}_{\text{SPG}}(f|p_0(h)) \).

We can train \( \hat{p}_{\text{SPG}}(f) \) on \( f_1, f_2, \ldots, f_m, \ldots, f_M \), by optimizing the aggregated ELBO on \( \hat{p}_{\text{SPG}}(f) \):

\[
\max_{\{w_s\}, \theta, \lambda} \mathbb{E}_{X_O} \sum_m \log \hat{p}_{\text{SPG}}(f_m^{X_O}) \geq \max_{\{w_s\}, \theta, \lambda} \mathbb{E}_{X_O} \sum_m \mathbb{E}_{\hat{q}_\lambda(h|f_m^{X_O})} \log \frac{\hat{p}_{\text{SPG}}(f_m^{X_O}|h)p_0(h)}{\hat{q}_\lambda(h|f_m^{X_O})},
\]

where \( f_m^{X_O} \) are the function values of \( f_m \) evaluated on \( X_O \). \( X_O \) are \(|O| \leq |D| \) measure points independently sampled from the training set \( X_D = \{x_i\}_{i=1}^N \). \( f_m^{X_O} \) are the function values of \( f_m \) evaluated on \( X_O \), and \( \hat{q}_\lambda(h|f_m^{X_O}) \) is an encoder network that approximates the true posterior \( \hat{p}_{\text{SPG}}(h|f_m^{X_O}) \).

**Product of Experts (PoE) encoder.** When sampling \( X_O \), since its size might vary each time, we would need to set up to \( 2^N \) inference nets, one for each possible subsets. To overcome this issue, we adopt the Product of Experts encoder [53], a simple and flexible approach for such scenario given by

\[
\hat{q}_\lambda(h|f_m(x_i), x_i) \propto p_0(h) \prod_i \hat{q}_\lambda(h|f_m(x_i), x_i),
\]

where \( \hat{q}_\lambda(h|f_m(x_i), x_i) \) is an inference network representing the expert associated with the \( i \)-th measurement point. Now the we can use a single encoder \( \hat{q}_\lambda(h|f_m^{X_O}) \) to handle all the possible inputs \( f_m^{X_O} \). In practice, we let \( \hat{q}_\lambda(h|f_m(x_i), x_i) \) to be a Gaussian expert that maps \([f_m(x_i), x_i]\) to a factorized Gaussian in latent space. Since the product of Gaussian experts is still Gaussian, \( \hat{q}_\lambda(h|f_m^{X_O}) \) is a Gaussian distribution whose statistics can be computed analytically.

**Estimating the grid-functional KL-divergence given \( X_n \).** In order to estimate the grid-functional KL divergence between \( q_{\text{SPG}}(f) \) and \( p(f) \), we first discuss how this divergence can be estimated on measurement points \( X_n \), i.e.,

\[
D_{KL}[q(f^{X_n})||p(f^{X_n})] \text{ where } f^{X_n} \text{ is the vector of function values}
\]
We denote the above variational family by $q$. Applying the approximation (17) to Equation 6, our final variational objective becomes

$$ f = \sum_s a_s \phi_s(\cdot, w_s) + \nu, \quad a \sim q(a) = \int p_0(a|h)q_0(h), \quad \nu \sim GP(\nu; 0, \sigma^2). \quad (15) $$

We denote the above variational family by $q_{SPG}(f|q_0(h))$. The key ingredient of our estimation method is that we force $q_{SPG}(f|q_0(h))$ and $\tilde{p}_{SPG}(f|p_0(h))$ to share the same basis functions (or weights $\{w_s\}$) and decoder parameters $\theta$. That is, once optimal $\{w_s\}$ and $\theta$ are obtained by fitting $\tilde{p}_{SPG}(f)$ to $p(f)$, these are frozen and reused in $q_{SPG}(f)$. This makes sense, since according to our definition in Section 3 [3], $q_{SPG}(f)$ and $\tilde{p}_{SPG}(f)$ share the same measurable space $(\mathbb{R}^T, \mathcal{B}_\mathbb{R}^T)$.

Therefore, the only difference between $q_{SPG}(f)$ and $\tilde{p}_{SPG}(f)$ is the choice of the prior distributions on $h$, which is $q_0(h)$ and $p_0(h)$, respectively. Given this property, we can compute the KL divergence between $q_{SPG}(f)$ and $\tilde{p}_{SPG}(f)$ given measurement points $X_n$ (Appendix A.5):

**Proposition 5 (KL divergence on measurement points between SPGs).** Let $q_{SPG}(f)$ and $\tilde{p}_{SPG}(f)$ be the SPGs defined in Equation 12 and 15. Then we have

$$ D_{KL}[q_{SPG}(f|X_n)||\tilde{p}_{SPG}(f|X_n)] = \mathbb{E}_{f \sim q_{SPG}(f)} \log Z(f|X_n), \quad (16) $$

where $Z(f|X_n)$ is the partition function, $Z(f|X_n) = \int h \tilde{p}_{SPG}(h|f|X_n) q_0(h) \frac{p_0(h)}{p_0(h)} dh$.

Note that $Z(f|X_n)$ is intractable to compute due to the intractability of the posterior $\tilde{p}_{SPG}(h|f|X_n)$. Fortunately, this is already approximated by the PoE inference net $\tilde{q}_\lambda(h|f|X_n)$ given by Equation 14:

$$ Z(f|X_n) \approx \tilde{Z}(f|X_n) := \int h \tilde{q}_\lambda(h|f|X_n) q_0(h) \frac{p_0(h)}{p_0(h)} dh. \quad (17) $$

Since $q_0(h)$, $p_0(h)$, and $\tilde{q}_\lambda(h|f|X_n)$ are all Gaussian distributions, $\tilde{Z}(f|X_n)$ can be computed using analytic solutions. Note also that thanks to the VAE-like structure in SPGs, all the calculations are performed in the latent space, whose dimensionality is much lower than $f$. With the additional help of analytic solutions for $\tilde{Z}(f|X_n)$, the estimation of (17) is very efficient and scalable.

### 4.4 The final algorithm: Mini-batching and de-biasing

So far we have been discussing the estimation of the KL-divergence on measurement points $X_n$. Next, we derive a practical algorithm based on the grid-functional ELBO $\mathcal{L}_{ELBO}$ defined in Equation 6. Applying the approximation (17) to Equation 6, our final variational objective becomes

$$ \log p(D) \geq \sum_{i=1}^{[D]} \mathbb{E}_{\eta(f)} [\log p_\eta(y_i | f(x_i))] - \mathbb{E}_{n, X_n \sim \mathcal{G}} D_{KL}[q(f|X_n)||p(f|X_n)] $$

$$ \approx \sum_{i=1}^{[D]} \mathbb{E}_{\eta(f)} [\log p_\eta(y_i | f(x_i))] - \mathbb{E}_{n, X_n \sim \mathcal{G}} \mathbb{E}_{f \sim \tilde{p}_{SPG}(f)} \log \tilde{Z}(f|X_n). $$

(18)

To make Equation 18 scalable to large data, we can apply mini-batch sampling to the likelihood term $\sum_{i=1}^{[D]} \mathbb{E}_{\eta(f)} [\log p_\eta(y_i | f(x_i))]$. Then the only bottleneck of Equation 18 is that the input $X_n$ is to the inference net $\tilde{q}_\lambda$ (used in $\tilde{Z}(f|X_n)$) can be very high dimensional due to the condition $X_D \subset X_n$ required by Proposition 3. Fortunately, we can derive the following mini-batch estimators for $\mathbb{E}_{n, X_n \sim \mathcal{G}} \mathbb{E}_{f \sim \tilde{p}_{SPG}(f)} \log \tilde{Z}(f|X_n)$ (Appendix A.6 and A.7):

**Proposition 6.** $E_{n, X_n \sim \mathcal{G}} \mathbb{E}_{f \sim \tilde{p}_{SPG}(f)} \log \tilde{Z}(f|X_n)$ can be estimated by the mini-batch estimator

$$ J_K := \frac{1}{2} \sum_{i=1}^{H} \mathbb{E}_{f \sim \tilde{p}_{SPG}(f)} \left[ \log \sigma_n^{-2} + \log (\hat{\sigma}_n^{-2} - \log (\sigma_n^{-2} + \hat{\sigma}_n^{-2} - 1) - \hat{\mu}_n^2 \hat{\sigma}_n^{-2} - \mu_n^2 \sigma_n^{-2} + (\hat{\sigma}_n^{-2} \hat{\mu}_n + \hat{\sigma}_n^{-2} \hat{\mu}_n)^2 (\sigma_n^{-2} + \hat{\sigma}_n^{-2} - 1)^{-1} \right], $$

where $\hat{\sigma}_n^2 = \sigma_n^2$ and $\hat{\mu}_n = \mu_n$.
Algorithm 1 Functional Variational Inference (FVI)

Require: data $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$; prior $p(f)$; surrogate $\tilde{q}_{\text{SPG}}(f)$, variational process $q_{\text{SPG}}(f)$, likelihood function $p_n(y_i | f)$, mini-batch sizes $I$ and $K$

1: while not converged do
2: Sample $[f_1(\cdot), f_2(\cdot), \ldots, f_M(\cdot)]$ from $p(f)$.
3: Improve $\tilde{q}_{\text{SPG}}(f)$ by optimizing the aggregated ELBO in Equation 13 w.r.t. $\{w_s\}, \theta, \lambda$.
4: end while
5: while not converged do
6: sample mini-batch $\mathcal{I}$ from $\{1, \ldots, |\mathcal{D}|\}$ and a set of measure points $X_n$ via c.
7: Optimize $\hat{L}_{\text{FVI}}$ in Equation 20 w.r.t. $\eta$ and $\pi$ via reparameterization tricks
8: end while

where $H$ is the dimensionality of $h$, $N(h; \mu_{h_i}, \sigma_{h_i}^2) = q_0(h_i), N(h; \mu_{\lambda_i}, \sigma_{\lambda_i}^2) = \tilde{q}_{\lambda}(h_i | f^X)$, $\hat{\sigma}_{\lambda_i}^2$ and $\hat{\mu}_{\lambda_i}$ are the mini-batch approximators for $\mu_{\lambda_i}$ and $\sigma_{\lambda_i}^2$, respectively:

$$\hat{\sigma}_{\lambda_i}^2 := \frac{1}{K} \sum_{k \in K} \sigma_{h_i | f^x_i}^2 + \sum_{x_i \in X_n \setminus X_D} \sigma_{h_i | f^x_i}^2,$$

$$\hat{\mu}_{\lambda_i} := \frac{1}{K} \sum_{k \in K} \sigma_{h_i | f^x_i}^{-2} + \sum_{x_i \in X_n \setminus X_D} \sigma_{h_i | f^x_i}^{-2} \mu_{h_i | f^x_i},$$

where $K$ is a mini-batch of size $K$ sampled from $\{1, \ldots, |\mathcal{D}|\}$, $x_i \in X_n \setminus X_D$ is a set of OOD samples sampled from $\mathcal{T}$ using c in Eq. 7 and $\mu_{h_i | f^x_i}$ and $\sigma_{h_i | f^x_i}^2$ are the mean and variance parameter returned from $\tilde{q}_{\lambda}(h_i | f^x_i))$.

The estimation in Eq. 19 is biased (but consistent). To remove the bias, we propose to debias Eq. 19 based on the Russian Roulette estimator [22]:

Proposition 7 (Debiasing). Let $R$ be a random integer from a distribution $\mathcal{P}(N)$ with support over the integers larger than $K$. $x_0$ is a random location sampled from $\mathcal{T}$. Then $\mathbb{E}_{n, x_0 \sim \mathcal{P}(N)} \mathbb{E}_{f \sim q_{\text{SPG}}(f)} \log \hat{Z}(f^x_n)$ can be estimated by $\mathbb{E} \left[ J_K + \sum_{k=K}^R \frac{\hat{\Delta}_k}{\mathcal{P}(N \geq k)} \right]$, where $\hat{\Delta}_k = \hat{J}_{k+1} - \hat{J}_k$, and the expectation is taken over $R, n, x_n, \pi$ and all mini-batches used by each $\hat{J}_k$ terms.

This will enable us to also perform mini-batch sampling on the measurement points when performing FVI. Our final optimization objective function is

$$\hat{L}_{\text{FVI}} := \frac{|\mathcal{D}|}{I} \sum_{i \in \mathcal{I}} \mathbb{E}_{q(f)}[\log p(y_i | f(x_i))] - \hat{J}_K - \sum_{k=B}^R \frac{\hat{\Delta}_k}{\mathcal{P}(N \geq k)}, \quad (20)$$

where $\mathcal{I}$ is a mini-batch of size $I$ for the likelihood terms, $R$ is an integer sampled from $\mathcal{P}(N)$, which is set to be $(R - K) \sim \text{Geom}(0.5)$. Finally, the full algorithm is sketched in Algorithm 1. We call this proposed method Functional Variational Inference (FVI).

Scalability. Our method is empirically much faster than f-BNN (Appendix C.3). When estimating $\mathcal{D}_{\text{grid}}[q_0(f) || p(f)]$, our method scales as $O(M_q)$, where $M_q$ is the number of samples sampled from $q_{\text{SPG}}(f)$ that are used in $\hat{J}_K$. In practice, we use $M_q = 1$. On the contrary, the SSG estimator used in f-BNN scales as $O(M_q^3 + M_q^2D)$, where much larger value needs to used (e.g., $M_q = 100$).

5 Related Works

Since BNNs and VIPs are discussed in Section 1 and 2, here we only address the rest of the related works including f-BNNs, functional priors, and Bayesian non-parametrics.

F-BNNs and F-PO. The functional BNN [47] is proposed to address the issue of specifying meaningful priors to BNN weights. It matches a BNN to a GP prior by minimizing the functional KL divergence estimated by score function estimators [27]. As discussed in [5], this objective is not well-defined for a wide class of distributions. Also, the score function estimators used in f-BNNs
often perform poorly in high dimensional spaces and are less efficient. In practice, we found that the f-BNN computational time is prohibitive. On the contrary, our FVI estimator avoids these issues by making use of the grid-functional divergence, which can be efficiently estimated using the latent representation of SPGs. More recently, the concurrent work of [39] proposes a tractable function space VI method for BNNs, in which the functional KL divergence is approximated via linearization. In their problem setting, the functional KL divergence is well defined since both $\mathcal{P}$ and $\mathcal{Q}$ are BNNs with the same neural network structures. Similarly, f-POVI [50] performs inference on BNN priors by performing particle optimization in function space. One limitation of f-POVI is that it requires the prior to be reparameterizable and differentiable while our method does not have this issue.

Function space priors. Another line of work directly defines distributions over functions by combining stochastic processes with neural networks. For example, neural processes (NPs) [13] and its variants [24, 15] focus on meta-learning scenarios and propose to use set encoders to model all possible posterior distributions of the form $\{p(f|C) | C \subset D\}$, where $C$ is the so-called “context points” in neural processes. This could be inefficient for large datasets, since it needs to feed all data points to the set encoder, which scales linearly w.r.t. the dataset size. More importantly, NPs still use for learning and inference an ELBO defined on parameter space instead of function space. On the contrary, our method focuses on the functional VI for supervised learning scenarios and does not need to model all possible conditionals. When computing predictive distribution, we only need to evaluate $q_\eta(h)$, which is a simple Gaussian distribution (no set encoders involved).

Bayesian non-parametrics. In the field of Bayesian non-parametrics, Gaussian Processes (GPs) [36] and deep GPs [8] are great examples of using function-space priors to produce models that are reliable under uncertainty. To reduce the prohibitive computation cost of exact/deep GPs, various VI methods [49, 17, 32, 40] have been studied. These methods share a similar principle with our work, that is, to minimize the functional divergence between the posterior and variational processes. Nevertheless, the GP components of the functional prior play a critical role in this line of work, which makes them less applicable to general non-GP based priors.

6 Experiments

In this section, we evaluate the performance of FVI using a number of tasks, including interpolation with structured implicit priors, multivariate regression with BNN priors, contextual bandits, and image classification. We mainly compare FVI with other weight-space and function-space Bayesian inference methods using the same priors. For more implementation details, please refer to Appendix B. Additional experiments can be found in Appendix C.

6.1 Interpolation with non-Gaussian priors: structured implicit priors

An advantage of FVI is that it can be applied to implicit (and non-Gaussian) priors over functions, where typical GPs do not apply. In this experiment, we evaluate the interpolation task as [47].
We compare the performance of FVI with ground truth, f-BNN and VIPs. The ground truth posterior is used as functional priors. We consider multivariate regression tasks based on 9 different UCI datasets. In this experiment, we test if the proposed FVI can perform accurate posterior inference with BNNs as functional priors; then, 100 observed data points are sampled as \( D \) half of which are sampled from \([0, 0.2]\) and the other half are sampled from \([0.8, 1]\). Finally, we ask the algorithms to perform inference using the prior, i.e., producing samples from \( p(f|D) \).

We compare the performance of FVI with ground truth, f-BNN and VIPs. The ground truth posterior samples are generated by sampling importance re-sampling. F-BNNs are based on the code kindly open-sourced by [47]. As we found that the training time required by f-BNN is prohibitive, we only trained f-BNN for 100 epochs for fairness (additional results for fully trained f-BNNs are provided in Appendix C.5). For VIPs (Gaussian approximations), we use an empirical covariance kernel, which is estimated from random function samples of the implicit priors. For FVI, implementation details can be found in Appendix B.2.

Results are displayed in Figure 3. FVI can successfully generate samples that mimic the piecewise constant/linear behaviors. The posterior uncertainty returned by FVI is also close to the ground truth estimates. On the other hand, f-BNNs severely under-fit the data and provide very poor in-between uncertainties. Note that, although f-BNNs are only trained for 100 epochs, their running time is still 100x higher than that of FVI (Appendix C.3). VIP performs better than f-BNNs, but fails to mimic the behaviour of the priors: the posterior samples from VIP are very noisy. This is due to the prior function samples violating the Gaussian assumption, with the correlation level between points being lower than expected. This results in very noisy VIP posterior samples that are hard to interpret.

### 6.2 Multivariate regression with BNNs priors

In this experiment, we test if the proposed FVI can perform accurate posterior inference with BNNs as functional priors. We consider multivariate regression tasks based on 9 different UCI datasets. We mainly compare with the following weight-space VI baselines for BNNs: Bayes-by-Backprop [4], variational dropout [12], and variational alpha dropout [20] (\( \alpha = 0.5 \)). We also compare with three function-space BNN inference methods: VIP-BNNs, VIP-Neural processes [28], and f-BNNs. Finally, we include comparisons to function space particle optimization [50] in Appendix C.7 for reference purpose. All inference methods are based on the same BNN priors whenever applicable. For experimental settings, we follow [28]. Each dataset was randomly split into train (90%) and test sets (10%). This was repeated 10 times and results were averaged.

Results are shown in Table 1. Overall, FVI consistently outperforms other VI-based inference methods for BNNs and achieves the best result in 7 datasets (out of 9). FVI also outperforms f-BNNs (in 5 datasets out of 6), despite the fact that they are more expensive to train. Note that exact GPs and

<table>
<thead>
<tr>
<th>DATASET</th>
<th>N</th>
<th>D</th>
<th>FVI</th>
<th>VIP-BNN</th>
<th>VIP-NP</th>
<th>BBB</th>
<th>VDO</th>
<th>( \alpha = 0.5 )</th>
<th>FBN</th>
<th>EXACT GP</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOSTON</td>
<td>100</td>
<td>14</td>
<td>2.53 ± 0.04</td>
<td>2.45 ± 0.04</td>
<td>2.43 ± 0.03</td>
<td>2.76 ± 0.04</td>
<td>2.67 ± 0.10</td>
<td>2.53 ± 0.04</td>
<td>2.10 ± 0.10</td>
<td>2.46 ± 0.04</td>
</tr>
<tr>
<td>CONCRETE</td>
<td>1030</td>
<td>20</td>
<td>3.88 ± 0.06</td>
<td>3.02 ± 0.02</td>
<td>3.13 ± 0.02</td>
<td>3.28 ± 0.01</td>
<td>3.23 ± 0.01</td>
<td>3.06 ± 0.03</td>
<td>3.09 ± 0.01</td>
<td>3.05 ± 0.02</td>
</tr>
<tr>
<td>ENERGY</td>
<td>768</td>
<td>8</td>
<td>0.5 ± 0.03</td>
<td>0.56 ± 0.04</td>
<td>0.6 ± 0.03</td>
<td>0.61 ± 0.02</td>
<td>0.11 ± 0.02</td>
<td>0.9 ± 0.01</td>
<td>0.68 ± 0.02</td>
<td>0.34 ± 0.02</td>
</tr>
<tr>
<td>KINSM</td>
<td>819</td>
<td>2</td>
<td>1.15 ± 0.01</td>
<td>1.12 ± 0.01</td>
<td>1.05 ± 0.00</td>
<td>0.81 ± 0.01</td>
<td>0.83 ± 0.01</td>
<td>0.92 ± 0.02</td>
<td>N/A ± 0.00</td>
<td>N/A ± 0.00</td>
</tr>
<tr>
<td>POWER</td>
<td>956</td>
<td>4</td>
<td>2.69 ± 0.00</td>
<td>2.92 ± 0.00</td>
<td>2.90 ± 0.00</td>
<td>2.83 ± 0.01</td>
<td>2.88 ± 0.00</td>
<td>2.81 ± 0.00</td>
<td>N/A ± 0.00</td>
<td>N/A ± 0.00</td>
</tr>
<tr>
<td>PROTEIN</td>
<td>1457</td>
<td>30</td>
<td>2.85 ± 0.00</td>
<td>2.87 ± 0.00</td>
<td>2.96 ± 0.02</td>
<td>3.01 ± 0.00</td>
<td>2.99 ± 0.00</td>
<td>2.90 ± 0.00</td>
<td>N/A ± 0.00</td>
<td>N/A ± 0.00</td>
</tr>
<tr>
<td>RED WINE</td>
<td>1588</td>
<td>11</td>
<td>0.97 ± 0.06</td>
<td>0.97 ± 0.02</td>
<td>1.0 ± 0.04</td>
<td>1.01 ± 0.02</td>
<td>0.97 ± 0.02</td>
<td>1.01 ± 0.02</td>
<td>1.04 ± 0.01</td>
<td>0.26 ± 0.03</td>
</tr>
<tr>
<td>YACHT</td>
<td>308</td>
<td>5</td>
<td>0.19 ± 0.01</td>
<td>0.02 ± 0.07</td>
<td>0.19 ± 0.01</td>
<td>0.07 ± 0.03</td>
<td>0.78 ± 0.01</td>
<td>0.79 ± 0.11</td>
<td>1.0 ± 0.03</td>
<td>0.10 ± 0.05</td>
</tr>
<tr>
<td>NAVAL</td>
<td>11594</td>
<td>14</td>
<td>7.21 ± 0.06</td>
<td>5.65 ± 0.04</td>
<td>11 ± 0.00</td>
<td>2.80 ± 0.00</td>
<td>2.80 ± 0.00</td>
<td>2.97 ± 0.14</td>
<td>7.13 ± 0.02</td>
<td>N/A ± 0.00</td>
</tr>
</tbody>
</table>

Table 2: Contextual bandits performance comparison. Results are relative to the cumulative regret of the worst algorithm on each dataset. Numbers after the algorithm are the network sizes. The best methods are boldfaced, and the second best methods are highlighted in brown.

<table>
<thead>
<tr>
<th>DATASET</th>
<th>N</th>
<th>D</th>
<th>FVI</th>
<th>VIP-BNN</th>
<th>VIP-NP</th>
<th>BBB</th>
<th>VDO</th>
<th>( \alpha = 0.5 )</th>
<th>FBN</th>
<th>EXACT GP</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAVAL</td>
<td>11594</td>
<td>14</td>
<td>7.21 ± 0.06</td>
<td>5.65 ± 0.04</td>
<td>11 ± 0.00</td>
<td>2.80 ± 0.00</td>
<td>2.80 ± 0.00</td>
<td>2.97 ± 0.14</td>
<td>7.13 ± 0.02</td>
<td>N/A ± 0.00</td>
</tr>
</tbody>
</table>

consider two 1-D implicit priors on \([0, 1]: 1\), piecewise constant random functions and 2), piecewise linear random functions. Please refer to Appendix B.2 for details. For each prior, we first sample a random function from the prior; then, 100 observed data points are sampled as \( D \), half of which are sampled from \([0, 0.2]\) and the other half are sampled from \([0.8, 1]\). Finally, we ask the algorithms to perform inference using the prior, i.e., producing samples from \( p(f|D) \).
Table 3: Image classification and OOD detection performance. Accuracy, negative log-likelihood (NLL) and area-under-the-curve (AUC) of OOD detection are reported. Our method outperforms all baselines in terms of classification accuracy and OOD-AUC and performs competitively on NLL for CIFAR10. Results for MAP, KFAC and Ritter et al. are obtained from [21].

<table>
<thead>
<tr>
<th>MODEL</th>
<th>FMNIST</th>
<th>CIFAR10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACCURACY</td>
<td>NLL</td>
</tr>
<tr>
<td>FVI</td>
<td>91.60 ± 0.14</td>
<td>0.254 ± 0.05</td>
</tr>
<tr>
<td>MFVI</td>
<td>91.20 ± 0.10</td>
<td>0.343 ± 0.01</td>
</tr>
<tr>
<td>MAP</td>
<td>91.39 ± 0.11</td>
<td>0.258 ± 0.00</td>
</tr>
<tr>
<td>KFAC-LAPLACE</td>
<td>84.42 ± 0.12</td>
<td>0.942 ± 0.01</td>
</tr>
<tr>
<td>RITTER ET AL</td>
<td>91.20 ± 0.07</td>
<td>0.265 ± 0.00</td>
</tr>
</tbody>
</table>

f-BNNs are not directly comparable to other methods, since i), they perform inference over different priors; and ii), they are much more expensive as they require the evaluation of the exact GP likelihood. Thus, their results are only available for smaller datasets, and are not included for ranking.

6.3 Contextual Bandits

Uncertainty estimates are important for downstream decision-making scenarios, since exploration-exploitation is a common dilemma that must be addressed. In this section, we consider a classic task called contextual bandits, where the agent is asked to make decisions that maximize the reward given some contexts (inputs). For this, Thompson sampling [48] is an elegant approach to guide exploration, where a model configuration is first sampled from the posterior and then an optimal action is chosen based on the sampled configuration.

We compare FVI with several NN-related baselines on datasets benchmarked by [37]. The hyperparameter settings are consistent with [47] except that we used a smaller batchsize (32). The learning rates for each baseline are tuned from [0.001, 0.05]. We report the cumulative regret as well as the mean ranks. Experiments are repeated for 10 runs. As shown in Table 2, no single algorithm always outperforms the others in all bandit problems. However, FVI tends to give better performance than the baselines (ranks the first overall, performs the best on 4 out of 7 datasets, and ranked top 2 on 6 out of 7 datasets), indicating that FVI can provide reliable uncertainty estimates for decision making. Moreover, FVI is much more efficient than f-BNN (nearly 500 times faster, c.f. Appendix C.4).

6.4 Image classification and out-of-distribution detection

To demonstrate the scalability of our method to higher dimensional data, we consider image classification tasks on Fashion MNIST [54] and CIFAR10 [25] with BNN priors. We compare our method to the following baselines: mean field VI (MFVI), maximum a posteriori (MAP), KFAC Laplace-GNN approximation [31] and its dampened version [38]. For all models, we use Bayesian CNNs with the same mixed CNN-fully connected structure as in [21, 43]. Apart from test accuracy and negative log likelihood (NLL), we also perform out-of-distribution detection using in-distribution (ID) / out-of-distribution (OOD) pairs including FashionMNIST/MNIST and CIFAR10/SVNH. Following the settings of [35, 21], we calculate the area under the curve (AUC) of out-of-distribution detection based on predictive entropies. Results are shown in Table 3. On both datasets, our proposed FVI method consistently outperforms all baselines in terms of (in-distribution) classification accuracy and OOD detection AUC. Although FVI does not achieve the best NLL on CIFAR10, it still performs competitively to MAP and dampened KFAC. This demonstrates that our method is able to scale to high dimensional data and produce accurate predictions with well-calibrated uncertainties.

7 Conclusion

In this paper, we propose Functional Variational Inference (FVI), a new VI method in function space. It optimizes a grid-based functional divergence, which can be estimated based on our proposed SPG model. We demonstrated that FVI works well with implicit priors, scales well to high dimensional data and provides reliable uncertainty estimates. Possible directions for future work might include developing grid-function KL estimation methods without surrogate models and improving the theoretical understanding of functional space VI.
Acknowledgements

We thank Javier Antoran and Yingzhen Li for helpful discussions and comments during the early stage of this work.

References


