Generalized Variational Inference in Function Spaces: Gaussian Measures meet Bayesian Deep Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

We develop a framework for generalized variational inference in infinite-dimensional function spaces and use it to construct a method termed Gaussian Wasserstein inference (GWI). GWI leverages the Wasserstein distance between Gaussian measures on the Hilbert space of square-integrable functions in order to determine a variational posterior using a tractable optimization criterion and avoids pathologies arising in standard variational function space inference. An exciting application of GWI is the ability to use deep neural networks in the variational parametrization of GWI, combining their superior predictive performance with the principled uncertainty quantification analogous to that of Gaussian processes. The proposed method obtains state-of-the-art performance on several benchmark datasets.

1 Introduction

In the past decade, considerable effort has been invested in developing Bayesian deep learning approaches [Welling and Teh, 2011, Chen et al., 2014, Blundell et al., 2015, Gal and Ghahramani, 2016, Kendall and Gal, 2017, Ritter et al., 2018, Khan et al., 2018, Maddox et al., 2019]. There are at least two key advantages to Bayesian models. Firstly, Bayesian model averaging is known to improve predictive performance [Komaki, 1996] even in misspecified situations [Fushiki, 2005, Ramamoorthi et al., 2015]. The empirical success of methods such as deep ensembles [Lakshminarayanan et al., 2017] may be interpreted as compelling evidence for this claim [Wilson and Izmailov, 2020]. Secondly, Bayesian models provide the user with a predictive distribution for an unseen data point. This can be naturally leveraged to quantify posterior uncertainty.

Even though impressive progress has been made, there are problems that remain unresolved. The prior distribution for the unknown function is typically induced by a prior distribution over deep neural network weights (and biases). It is hard to interpret the inductive bias in a function space that is induced by such priors for weights and unclear how one might incorporate prior knowledge about the unknown function. Additionally, the resulting inference problem is extremely high-dimensional and requires approximation techniques that are either computationally expensive [Neal, 2012] or so crude that the approximate posterior may suffer from pathological behavior [Foong et al., 2020]. The difficulties of performing Bayesian inference for weights have led to the emergence of methods that approach the problem in function space directly [Ma et al., 2019, Sun et al., 2019, Rudner et al., 2020, Ma and Hernández-Lobato, 2021].

The theory of constructing prior distributions in function spaces is well developed and the most famous class of prior distributions are *Gaussian processes*. They have been commonly used for

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

decades in the machine learning community to elicit interpretable functional priors and are known to have well-calibrated predictive uncertainties [Rasmussen, 2003].

In a separate thread of research, a new powerful inference framework called *Generalized Variational Inference* (GVI) has been recently developed [Knoblauch et al., 2019]. The authors argue that standard assumptions of Bayesian inference such as well-specified priors, well-specified likelihoods and infinite computing power are often violated in practice. They therefore propose a generalized view on Bayesian inference that takes these points into consideration. We extend the work of Knoblauch et al. [2019] to situations where no probability density functions for the prior exist and are thus able to use generalized variational inference in infinite-dimensional function spaces directly. We then specify both the prior and variational measures as Gaussian measures and measure their dissimilarity using the Wasserstein distance. This results in the method which we call *Gaussian Wasserstein Inference in Function Spaces* (GWI-FS). An exciting application of our method is the ability to equip deep neural networks with uncertainty quantification using the framework analogous to that of Gaussian processes, resulting in a state-of-the-art method termed *GWI-net*. Our main contributions are:

- We create a general framework for inference in function space based on Gaussian measures on the space of square-integrable functions,
- We derive an objective function that can be expressed in terms of the *parameters of the Gaussian measures*,
- We derive a tractable approximation to our objective function that is valid for (almost) arbitrary kernels and mean functions,
- We demonstrate the utility of our method by obtaining state-of-the-art results on the UCI regression datasets and on Fashion MNIST and CIFAR 10 [1].

2 Related Work

GWI-FS draws on the work developed in the Gaussian process literature, but can be used to equip traditional neural network architectures with uncertainty. We therefore give a brief overview of the relevant related methods in both the Bayesian neural network (BNNs) and Gaussian process community.

Bayesian neural networks Traditionally Bayesian neural networks have been assigned priors in weight space. The effects of various priors on inference and uncertainty quantification are still not well understood [Fortuin et al., 2021]. As the posterior (over weights) is intractable, sampling algorithms such as Hamiltonian Monte Carlo (HMC) were initially proposed Neal [2012]. Due to the unfavorable scaling properties of standard HMC which requires the full gradient, batch-size approximations of HMC evolved [Chen et al., 2014]. Another line of research exploits Langevin dynamics to generate posterior samples [Welling and Teh, 2011] in weight space.

Variational methods for BNNs in weight space In variational inference, the true posterior is approximated by a more tractable so-called *variational* distribution. The user specifies a class of approximate posterior measures and selects the best posterior approximation by maximizing the so-called evidence lower bound (ELBO). The Bayes by Backprop [Blundell et al., 2015] method is one such variational mean-field approximation of the weight-space posterior. In variational dropout [Gal and Ghahramani, 2016], a specific approximation is chosen to reinterpret dropout [Srivastava et al., 2014] at test time as a variational procedure.

Variational methods for BNNs in function spaces Inference in weight space is challenging, as the problem is typically high-dimensional and the posterior distribution over weights multi-modal. This led to a line of research in which inference algorithms are formulated in function spaces. Variational implicit processes [Ma et al., 2019] approximate the BNN posterior as a linear combination of draws from the prior. Functional-BNN [Sun et al., 2019] matches a BNN to a GP prior and performs inference by optimising a functional Kullback-Leibler (KL) divergence exploiting score function estimators [Li and Turner, 2017]. Rudner et al. [2020] use a local approximation to the prior and variational posterior processes to obtain a tractable functional Kullback-Leibler divergence. Ma and

[1] Codebase: https://anonymous.4open.science/r/GWI-D7CA/

Hernández-Lobato [2021] generalise the variational family in Ma et al. [2019] and obtain a more scalable procedure by using a different approximation to the functional KL-divergence. Recent work has also proposed to adapt BNN priors to interpretable functional priors by minimizing the Wasserstein distance between a BNN prior and a Gaussian process [Tran et al., 2020].

Gaussian processes Standard Gaussian process regression [Rasmussen, 2003] allows interpretable prior specification but scales poorly with respect to the number of data points. As a result, a plethora of approximation techniques are introduced. On one hand, there are variational approximations to the true posterior [Titsias, 2009, Hensman et al., 2013] and several extensions [Hensman et al., 2017, Salimbeni et al., 2018, Dutordoir et al., 2020]. On the other hand, GPU utilization is combined with Krylov subspace methods to obtain scalability [Gardner et al., 2018, Wang et al., 2019].

3 Background

In this section we give some background on generalized variational inference in infinite dimensions and introduce Gaussian measures in Hilbert spaces. We further discuss their relation to the more familiar Gaussian processes at the end.

3.1 Generalized Variational Inference in Function Spaces

In functional variational inference, we assign a prior $p(f)$ to the unknown function $f \in E$, where E is a function space[2]. The prior is combined with the likelihood $p(y|f)$ to give the posterior $p(f|y)$. The posterior is often intractable which is why in variational inference we specify a tractable variational approximation $q(f)$ to $p(f|y)$ and train our model by maximising the evidence lower bound (ELBO)

$$\mathcal{L} = \mathbb{E}_{q(f)}\big[\log p(y|f)\big] - \mathbb{D}_{\mathrm{KL}}\big(q(f), p(f)\big), \tag{1}$$

where $\mathbb{D}_{\mathrm{KL}}$ denotes the KL divergence. Note that in the case where E is infinite dimensional $p(f)$ and $q(f)$ cannot be probability density functions with respect to the Lebesgue measure [see e.g. Hunt et al., 1992, for a discussion], which is why the above notation, although commonly used, is imprecise. What we in fact mean are the probability measures over E associated with the prior and variational approximation. We will denote these measures as $\mathbb{P}^F$ and $\mathbb{Q}^F$ from now on to make this difference explicit. The ELBO in this notation reads as

$$\mathcal{L} := \mathbb{E}_{\mathbb{Q}}\big[\log p(y|F)\big] - \mathbb{D}_{\mathrm{KL}}\big(\mathbb{Q}^F, \mathbb{P}^F\big). \tag{2}$$

Note that the KL divergence (for measures) is defined as

$$\mathbb{D}_{\mathrm{KL}}\big(\mathbb{Q}^F, \mathbb{P}^F\big) = \int \log\left(\frac{d\mathbb{Q}^F}{d\mathbb{P}^F}(f)\right) d\mathbb{Q}^F(f), \tag{3}$$

where we assume that $\mathbb{Q}^F$ is dominated by the measure $\mathbb{P}^F$ which guarantees the existence of the Radon-Nikodym derivative $d\mathbb{Q}^F/d\mathbb{P}^F$. A number of papers focus on obtaining tractable approximations of (3) [Sun et al., 2019, Rudner et al., 2020, Ma and Hernández-Lobato, 2021]. However, the use of KL-divergence in infinite-dimensional function spaces can be a delicate task, since benign constructions of priors and variational approximations may not satisfy that $\mathbb{Q}^F$ is dominated by $\mathbb{P}^F$ which leads to $\mathbb{D}_{\mathrm{KL}}\big(\mathbb{Q}^F, \mathbb{P}^F\big) = \infty$ [Burt et al., 2020]. This often renders the objective (2) useless or at least problematic.

A *true Bayesian* is committed to the use of the KL divergence in (2) as maximizing $\mathcal{L}$ is equivalent to minimizing the KL divergence between the true posterior measure and the variational measure. This equivalence is typically demonstrated using pdfs but the argument generalizes to infinite dimensions as is shown for GPs in Matthews et al. [2016] or in a more measure theoretic formulation in Theorem 4 of Wild and Wynne [2021].

However, Knoblauch et al. [2019] argue that given the problems of prior and likelihood specification as well as available compute, an axiomatically justified way of moving from prior to posterior beliefs is by solving a more general optimization problem [Knoblauch et al., 2019, Theorem 15].

[2] We assume E to be a Polish space, which avoids technical difficulties in defining the posterior measure [Ghosal and Van der Vaart, 2017, Chapter 1.3]

Crucially it is valid to replace the KL-divergence by an arbitrary measure of dissimilarity $\mathbb{D}$ satisfying $\mathbb{D}(\mathbb{Q}^F, \mathbb{P}^F) \geq 0$ and $\mathbb{D}(\mathbb{Q}^F, \mathbb{P}^F) = 0 \Rightarrow \mathbb{Q}^F = \mathbb{P}^F$. The arguments in Knoblauch et al. [2019] are made assuming the existence of a pdf for the prior, but they rely solely on a reformulation of Bayesian inference as optimization problem [Knoblauch et al., 2019, Chapter 2]. We show in Appendix A.1 that this reformulation can also be made for infinite-dimensional prior measures and therefore consider the generalized loss

$$\mathcal{L} := -\mathbb{E}_{\mathbb{Q}}\big[\log p(y|F)\big] + \mathbb{D}\big(\mathbb{Q}^F, \mathbb{P}^F\big), \tag{4}$$

a valid optimization objective for an arbitrary dissimilarity measure $\mathbb{D}$. This is merely an infinite-dimensional version of equation (10) in Knoblauch et al. [2019]. We refer to inference targeting the objective (4) as *Generalized variational inference in function space* (GVI-FS).

Henceforth, the particular instance of GVI-FS that we explore is where both $\mathbb{P}^F$ and $\mathbb{Q}^F$ are Gaussian measures (on an infinite-dimensional Hilbert space) and $\mathbb{D}$ is chosen to be the Wasserstein metric [Kantorovich, 1960]. We will refer to this setting as *Gaussian Wasserstein Inference in Function Space* (GWI-FS).

3.2 Gaussian Random Elements and Gaussian Measures in Hilbert spaces

In this section we introduce Gaussian random elements (GRE) and Gaussian measures in Hilbert spaces – these concepts are somewhat technical but crucial in the construction of our method. We then describe their close relationship to the more familiar Gaussian process notions in the next section.

Let $\big(\Omega, \mathcal{A}, \mathbb{P}\big)$ be the underlying (physical) probability space and $\big(H, \langle\cdot,\cdot\rangle\big)$ be a Hilbert space.

Gaussian random elements A measurable function $F : \Omega \to H$ is called GRE (in H) if and only if $\langle F, h\rangle : \Omega \to \mathbb{R}$ has a scalar Gaussian distribution for all $h \in H$.[3] Every GRE F has a mean element $m \in H$ defined by

$$m := \int F(\omega)\, d\mathbb{P}(\omega) \tag{5}$$

and a (linear) covariance operator $C : H \to H$ defined by

$$Ch(\cdot) := \int \langle F(\omega), h\rangle F(\omega)\, \mathbb{P}(\omega) - \langle m, h\rangle m. \tag{6}$$

for $h \in H$. Both integrals are to be understood as Bochner integrals [Kukush, 2020, Chapter 3]. The Bochner integral has the property that $\big\langle \int F(\omega)\, d\mathbb{P}(\omega), h\big\rangle = \int \langle F(\omega), h\rangle\, d\mathbb{P}(\omega)$ for all $h \in H$. This combined with Fubini's theorem and the definition of a GRE implies that

$$\langle F, h\rangle \sim \mathcal{N}\big(\langle m, h\rangle, \langle Ch, h\rangle\big), \tag{7}$$

for any $h \in H$ with $\mathcal{N}(\mu, \sigma^2)$ denoting the normal distribution with mean $\mu \in \mathbb{R}$ and variance $\sigma^2 > 0$. Similarly we denote $F \sim \mathcal{N}(m, C)$ for a GRE in H with mean element m and covariance operator C. It can be shown that the covariance operator C of a GRE is a positive self-adjoint trace-class operator. Conversely, for every positive self-adjoint trace class operator and every $m \in H$, there exists a GRE with $F \sim \mathcal{N}(m, C)$ [Bogachev, 1998, Theorem 2.3.1].

Gaussian measures The push-forward measure of $\mathbb{P}$ through F is defined as $\mathbb{P}^F(A) := \mathbb{P}\big(F^{-1}(A)\big)$ for all Borel-measurable $A \subset H$. If $F \sim \mathcal{N}(m, C)$ is a GRE, we call $P := \mathbb{P}^F$ a GM and write $P = \mathcal{N}(m, C)$. Note that GMs or equivalently GREs allow us to specify probability distributions over (infinite-dimensional) Hilbert spaces by using a given mean element and a given covariance operator.

Details about Gaussian Measures in Hilbert spaces can be found in Chapter 2 of Da Prato and Zabczyk [2014] or in Kukush [2020]. In fact, Gaussian measures can be defined on even more general linear spaces such as Banach or Fréchet spaces [Bogachev, 1998].

[3] We allow for the degenerate case where the variance of $\langle F, h\rangle$ is zero. This means we interpret a Gaussian with variance zero as Dirac measure.

3.3 Gaussian Processes and Their Corresponding Measures

In this section we describe how Gaussian processes – a standard tool to assign functional priors in Bayesian machine learning – are related to Gaussian measures.

Let $\big(\Omega, \mathcal{A}, \mathbb{P}\big)$ be the underlying (physical) probability space and $\mathcal{X} \subset \mathbb{R}^D$ be measurable. The (product-) measurable mapping $G : \Omega \times \mathcal{X} \to \mathbb{R}$ is called a Gaussian process (GP) if and only if for all $N \in \mathbb{N}$ and all $X = \{x_n\}_{n=1}^N \subset \mathcal{X}$ the random vector $G(X) := \big(G(\cdot, x_1), \ldots, G(\cdot, x_N)\big)^T$ is multivariate Gaussian. For a GP G we define a mean function $m(x) := \mathbb{E}\big[G(x)\big]$, $x \in \mathcal{X}$, and a covariance function by $k(x, x') := \mathbb{C}[G(x), G(x')\big]$ for $x, x' \in \mathcal{X}$. Here $\mathbb{E}$ denotes the expected value and $\mathbb{C}[\cdot, \cdot]$ the covariance. It follows from the definition that $G(X) \sim \mathcal{N}\big(m(X), k(X, X)\big)$ for any $\{x_n\}_{n=1}^N \subset \mathcal{X}$, where we define $m(X) := \big(m(x_n)\big)_{n=1}^N$ and $k(X, X) := \big(k(x_n, x_{n'})\big)_{n,n'=1}^N$. We write $G \sim GP(m, k)$ for a GP with mean function m and covariance function k. Note that by the properties of the covariance we know that $k(X, X)$ is a (symmetric) positive semi-definite matrix for all $\{x_n\}_{n=1}^N \subset \mathcal{X}$ and $N \in \mathbb{N}$. A function with this property is called *kernel*, a terminology that we adopt henceforth. Kolmogorov's existence theorem [Billingsley, 2008, Section 36] guarantees the existence of a Gaussian process for any kernel k and any mean function m. The standard reference for Gaussian processes in machine learning is Rasmussen [2003].

The main advantage of Gaussian processes in specifying priors over a function space is that the kernel k allows us to incorporate readily interpretable prior assumptions, such as smoothness or periodicity. For example, choosing the squared exponential kernel [Rasmussen, 2003] implies that the unknown function is infinitely differentiable and that the correlation of the functional output is higher the closer the inputs are.

In order to insert the Gaussian process prior into our generalized loss in (4) we need to know the probability measure that is associated to the Gaussian process. In general, we can associate more than one Gaussian measure with a given Gaussian process. For example:

- If the GP has continuous sample paths we can associate a Gaussian measure on the space E of continuous functions with it [Lifshits, 2012, Example 2.4].
- If the GP has square-integrable sample paths we can associate a Gaussian measure on the Hilbert space of square-integrable functions with it (cf. Theorem 1).

These sample path properties can be guaranteed under additional assumptions on the kernel. The next theorem discusses one such kernel condition which guarantees the GP to have sample paths in the Hilbert space of square integrable functions, denoted $L^2(\mathcal{X}, \rho, \mathbb{R})$, with inner product $\langle g, h\rangle_2 := \int_{\mathcal{X}} g(x)h(x)\, d\rho(x)$.

Theorem 1. *Let $F \sim GP(m, k)$ be a GP with mean $m \in L^2(\mathcal{X}, \rho, \mathbb{R})$ and kernel k such that*

$$\int_{\mathcal{X}} k(x, x)\, d\rho(x) < \infty. \tag{8}$$

We call a kernel satisfying (8) *trace-class kernel. Then the mapping $\widetilde{F} : \Omega \to L^2(\mathcal{X}, \rho, \mathbb{R})$ defined as $\widetilde{F}(\omega) := F(\omega, \cdot)$ is a Gaussian random element with mean m and covariance operator C given as*

$$Cg(\cdot) := \int k(\cdot, x')g(x')\, d\rho(x') \tag{9}$$

for any $g \in L^2(\mathcal{X}, \rho, \mathbb{R})$. Consequently $P := \mathbb{P}^F \sim \mathcal{N}(m, C)$ is a Gaussian measure.

Proof. The fact that $\widetilde{F}$ as defined above is a GRE follows immediately from Example 2.3.16 in Bogachev [1998]. The fact that m is its mean and C as defined in (9) is its covariance operator follows from Fubini's theorem. □

It shall be noted that there is no need to appeal to GPs in order to justify the use of GMs. In fact, it has recently been demonstrated that variational inference for GPs can be formulated purely in terms of GMs [Wild and Wynne, 2021]. In the following sections we will therefore deploy GMs without any reference to GPs, but it is of course always possible to think of them as the measures that correspond to GPs where the kernel satisfies an additional assumption such as (8).

4 Gaussian Wasserstein Inference in Function Spaces

This section describes how the Wasserstein distance between Gaussian measures can be used to obtain a tractable optimization target for inference in function spaces. In the end, we discuss several parametrizations of GWI and introduce our main inference method - the GWI-net.

4.1 Model description

Let $\{(x_n, y_n)\}_{n=1}^N \subset \mathcal{X} \times \mathcal{Y}$ be $N \in \mathbb{N}$ paired observations. We assume that $\mathcal{X} \subset \mathbb{R}^D$, $D \in \mathbb{N}$ and further that $\mathcal{Y} = \mathbb{R}$ for regression and $\mathcal{Y} = \{1, \dots, J\}$ for classification with $J \in \mathbb{N}$ classes. We focus in our exposition here on the regression case but have given the relevant derivations for classification in Appendix A.6.

As pointed out in section 3.1, GVI in function space minimises the generalized loss $\mathcal{L} = -\mathbb{E}_{\mathbb{Q}}\big[\log p(y|F)\big] + \mathbb{D}\big(\mathbb{Q}^F, \mathbb{P}^F\big)$. We make the mild assumption that the unknown function f is square integrable with respect to the data distribution ρ on $\mathcal{X}$ which means $f \in E = L^2(\mathcal{X}, \rho, \mathbb{R})$. The prior $P := \mathbb{P}^F$ is described by a Gaussian measure with mean $m_P \in \mathcal{L}^2(\mathcal{X}, \rho, \mathbb{R})$ and covariance operator C_P described by a trace-class kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which means it is given as $(C_P f)(x) := \int_{\mathcal{X}} k(x, x') f(x')\, d\rho(x')$ for all $f \in L^2(\mathcal{X}, \rho, \mathbb{R})$. We assume a Gaussian likelihood for $y := (y_1, \dots, y_N)$ given as $p(y|f) := \prod_{n=1}^N p(y_n|f)$[4] with

$$p(y_n|f) := \mathcal{N}(y_n \mid f(x_n), \sigma^2), \tag{10}$$

where $\mathcal{N}(\cdot \mid \mu, \sigma^2)$ denotes the pdf of a normal distribution with mean $\mu \in \mathbb{R}$ and variance $\sigma^2 > 0$. This prior and likelihood are natural choices as they mimic the standard formulation of Gaussian process regression. The variational approximation of the posterior is chosen to be another Gaussian measure $Q := \mathbb{Q}^F$ with arbitrary mean $m_Q \in L^2(\mathcal{X}, \rho, \mathbb{R})$ and arbitrary covariance operator C_Q induced by a trace-class kernel r: $(C_Q f)(x) := \int_{\mathcal{X}} r(x, x') f(x')\, d\rho(x')$ for all $f \in L^2(\mathcal{X}, \rho, \mathbb{R})$.

It remains for us to select a dissimilarity measure $\mathbb{D}$. As already pointed out in the introduction we decide to use the Wasserstein distance W_2 (a formal definition is given in Appendix A.3). This choice was guided by two considerations:

1. The Wasserstein metric was proven to be a useful metric for probability distributions in machine learning applications [Arjovsky et al., 2017, Tran et al., 2020]. Furthermore the Wasserstein metric is known to have desirable statistical properties [Panaretos and Zemel, 2019].
2. The Wasserstein distance is tractable for arbitrary Gaussian measures on (separable) Hilbert spaces [Gelbrich, 1990] and given as

$$W_2^2(P, Q) = \|m_P - m_Q\|_2^2 + tr(C_P) + tr(C_Q) - 2 \cdot tr\Big[\big(C_P^{1/2} C_Q C_P^{1/2}\big)^{1/2}\Big], \tag{11}$$

   where tr denotes the trace of an operator and $C_P^{1/2}$ is the square root of the positive, self-adjoint operator C_P. This is in stark contrast to the KL-divergence that is infinite whenever $\mathbb{Q}^F$ is not dominated by $\mathbb{P}^F$ and even in the case where it is finite there exists no explicit formula for the KL-divergence in infinite dimensions.

The generalized loss for our model is therefore given as

$$\mathcal{L} = -\sum_{n=1}^N \mathbb{E}_{\mathbb{Q}}\Big[\log \mathcal{N}\big(y_n \mid F(x_n), \sigma^2\big)\Big] + W_2(P, Q). \tag{12}$$

Note that the expected log-likelihood in (12) can be calculated analytically as

$$\mathbb{E}_{\mathbb{Q}}\Big[\log \mathcal{N}\big(y_n \mid F(x_n), \sigma^2\big)\Big] = -\frac{N}{2}\log(2\pi\sigma^2) - \sum_{n=1}^N \frac{\big(y_n - m_Q(x_n)\big)^2 + r(x_n, x_n)}{2\sigma^2}. \tag{13}$$

[4] Astute readers may notice that the definition of the likelihood contains a pointwise evaluation $f(x_n)$ which may not be a well defined operation on $L^2(\mathcal{X}, \rho, \mathbb{R})$. We detail in Appendix 30 how that problem can be circumvented and that in fact $F(x) \sim \mathcal{N}(m(x), k(x, x))$ as one would expected.

It remains to produce an approximation of (11) in order to obtain a tractable inference procedure. To this end, note that by definition $\|m_P - m_Q|_2^2 = \int \big(m_P(x) - m_Q(x)\big)^2 \, d\rho(x)$ and further $tr(C_P) = \int k(x,x)\, d\rho(x)$ [Brislawn, 1991]. We now replace the true input distribution ρ with the empirical data distribution $\widehat{\rho} := \frac{1}{N}\sum_{n=1}^N \delta_{x_n}$, where δ_x denotes the Dirac measure in $x \in \mathcal{X}$. This gives $\|m_P - m_Q\|_2^2 \approx \frac{1}{N}\sum_{n=1}^N \big(m_P(x_n) - m_Q(x_n)\big)^2$, $tr(C_P) \approx \frac{1}{N}\sum_{n=1}^N k(x_n,x_n)$ and $tr(C_Q) \approx \frac{1}{N}\sum_{n=1}^N r(x_n,x_n)$. It remains to provide an approximation of $tr\Big[\big(C_P^{1/2} C_Q C_P^{1/2}\big)^{1/2}\Big]$. The key idea is to approximate the spectrum of $C_P^{1/2} C_Q C_P^{1/2}$ by that of an appropriate kernel matrix. Details are discussed in Appendix A.4. This leads to the following final approximation for the Wasserstein metric

$$\hat{W} := \frac{1}{N}\sum_{n=1}^N \big(m_P(x_n) - m_Q(x_n)\big)^2 + \frac{1}{N}\sum_{n=1}^N k(x_n,x_n) \tag{14}$$

$$+ \frac{1}{N}\sum_{n=1}^N r(x_n,x_n) - \frac{2}{\sqrt{N N_S}} \sum_{s=1}^{N_S} \sqrt{\lambda_s\big(r(X_S,X)k(X,X_S)\big)}, \tag{15}$$

where $X_S := (x_{S,1},\dots,x_{S,N_S})$ with $x_{S,1},\dots x_{S,N_S} \in \mathbb{R}^D$ being subsampled from the input data X. Further $r(X_S,X) := \big(r(x_{S,s},x_n)\big)_{s,n}$ and $k(X,X_S) := \big(k(x_n,x_{S,s})\big)_{n,s}$ for $n = 1,\dots,N$, $s = 1,\dots,N_S$ and $\lambda_s\big(r(X_S,X)k(X,X_S)\big)$ denotes the s-th eigenvalue of the matrix $r(X_S,X)k(X,X_S) \in \mathbb{R}^{N_S \times N_S}$.

The combination of (13), (14) and (15) gives a generalized loss that is tractable in terms of m_P, m_Q, k, and r. If we disregard computation time of m_P, m_Q, k and r, the generalized loss can be evaluated in $\mathcal{O}(N + N_S^2 N + N_S^3)$, where typically $N_S \ll N$, e.g. $N_S = 100$. We provide a batch version of our loss in Appendix A.5 which reduces the computations to $\mathcal{O}(N_S^2 N_B + N_S^3)$ where $N_B \ll N$ is the batch-size. Note, however, that the final computation time for our method will be determined by the complexity hidden in the evaluation of m_Q, m_P, k, and r as we need N_B evaluations of m_Q and m_P and $N_S \cdot N_B$ evaluations of r and k per iteration.

4.2 Parameterisations of Prior and Variational Measure

The prior for our model is given as $P = \mathcal{N}(m_P, C_P)$ with C_P induced by a trace-class kernel k. One of the advantages of the proposed approach is that any trace-class kernel is allowed and this is where one can incorporate specific assumptions and domain expertise. This is a thoroughly studied topic: the prior kernel can encode periodicity [Durrande et al., 2016], geometric intuition [van der Wilk et al., 2018], and even model linear constraints for the unknown function [Jidling et al., 2017]. In order to keep the exposition simple and maintain focus on the inference, however, and in line with using simple priors on network weights in standard Bayesian deep learning, we opt for a simple zero mean prior $m_P = 0$ and a standard ARD kernel k given as

$$k(x,x') = \sigma_f^2 \exp\Big(-\frac{1}{2}\sum_{d=1}^D \frac{(x_d - x'_d)^2}{\alpha_d^2}\Big) \tag{16}$$

for $x, x' \in \mathcal{X} \subset \mathbb{R}^D$. We refer to $\sigma_f > 0$ as *kernel scaling factor* and to $\alpha_d > 0$ as *length-scale* for dimension d. The parameters σ_f and $\alpha := (\alpha_1,\dots,\alpha_D)$ are called *prior hyperparameters*.

The rest of the section explores various choices for the variational mean m_Q and the variational kernel r. The parameters appearing in the specification of m_Q and r are referred to as *variational parameters*.

GWI: Stochastic variational Gaussian process Let $z_1,\dots,z_M \in \mathcal{X}$ be a subsample of the data X with $M \ll N$. We define the posterior mean

$$m_Q(x) := m_P(x) + \sum_{m=1}^M \beta_m k_m(x) \tag{17}$$

with $\beta_m \in \mathbb{R}$ and $k_m(x) := k(x, z_m)$, $m = 1, \dots, M$ where k is the prior kernel k and $\beta := (\beta_1, \dots, \beta_M) \in \mathbb{R}^M$ are variational parameters. Define further the variational kernel

$$r(x, x') = k(x, x') - k_Z(x)^T k(Z, Z)^{-1} k_Z(x) + k_Z(x)^T \Sigma k_Z(x). \tag{18}$$

This choice of m_Q and r essentially recovers the *stochastic variational Gaussian processes* (SVGP) model of Titsias [2009]. Note that in our framework it is straightforward to use all (or just more) basis functions for the mean $m_Q(x) := m_P(x) + \sum_{n=1}^{N} \beta_n k_n(x)$ where $k_n(x) := k(x, x_n)$, $\beta_n \in \mathbb{R}$, $n = 1, \dots, N$. This mirrors the construction in Cheng and Boots [2017] where we allow more parameters to learn the mean than in SVGP. However, both Titsias [2009] and Cheng and Boots [2017] use a different objective function than GWI to learn the unknown parameters.

GWI: deep neural network with SVGP An interesting approach is to parameterise the posterior mean as a deep neural network (DNN). We assume the DNN has $L \in \mathbb{N}$ hidden layers and the width of layer $\ell = 1, \dots, L$ is denoted D_ℓ with $D_0 := D$ and $D_{L+1} = 1$. This means we define $g^1(x) := W^1 x + b^1$ and further $h^\ell(x) := \phi\big(g^\ell(x)\big)$, $g^{\ell+1}(x) := W^{\ell+1} h^\ell(x) + b^{\ell+1}$ for $\ell = 1, \dots, L$. Here W^{l+1} is $D_{\ell+1} \times D_\ell$ matrix, $b^{\ell+1} \in \mathbb{R}^{D_{\ell+1}}$ is a bias vector for layer l and ϕ an activation function. We can then define the variational mean as $m_Q(x) := m_P(x) + g^{L+1}(x)$. If we choose the SVGP kernel r in (18), we essentially predict with a neural network and quantify uncertainty with a (sparse) Gaussian process, capturing the beneficial properties of both.

Neural networks have been combined in several ways with GPs [Wilson et al., 2016, Tran et al., 2020]. However, to the best of our knowledge they were not used to directly parametrize the posterior in the context of generalized variational inference in function space. The spirit of our approach is fundamentally different: rather than thinking of a neural network as a model which needs to be made Bayesian, we use it as a parametrisation of a variational posterior.

We note that we do not here provide an exhaustive study on how to best parameterize the variational measure. This paper is focused on demonstrating the ability of the proposed method to obtain valid uncertainty quantification. An exploratory study on how properties and quality of uncertainty quantification relate to different choices of m_Q and r is reserved for future work. We mention potential problems that can occur from misspecification in Appendix A.10.

5 Experiments

We show results for GWI with the SVGP mean (17) and the SVGP kernel (18). We use the shorthand GWI: SVGP for this approach. Additionally we implement the DNN mean with the SVGP kernel (18). This combination achieves impressive results on various regression and classification tasks. We call this method GWI: DNN-SVGP or simply GWI-net.

Illustrative Examples In Figure 1 we illustrate GWI-net on a few toy examples. One can clearly see that the posterior predictive variance expands for regions lacking observations which demonstrates the ability of our method to quantify uncertainty. Additionally, we provide an example for two-dimensional inputs in Appendix A.9. There we show that the pathologies regarding the quantification of in-between uncertainty discussed in Foong et al. [2020] are not present for our method.

UCI Regression In Table 1 we report the average test negative log-likelihood (NLL) (cf. Appendix A.7 for a definition) of GWI: SVGP and GWI-net (GWI: DNN-SVGP) and the results of several weight-space approaches for BNNs: Bayes-by-Backprop (BBB) [Blundell et al., 2015], variational dropout (VDO) [Gal and Ghahramani, 2016], and variational alpha dropout ($\alpha = 0.5$) [Li and Gal, 2017]. We also compare with four function-space BNN inference methods: functional variational inference with BNN prior (FVI) [Ma and Hernández-Lobato, 2021], variationally implicit processes (VIP) with BNNs, VIP-Neural processes [Ma et al., 2019], and functional BNNs (FBNNs) [Sun et al., 2019].

One can see that GWI-net obtains the best mean rank of all methods being the best model on 4/9 datasets and performing competitively on all datasets. Note that we exclude FBNN and exact Gaussian processes from the comparison because their computational complexity is often prohibitively large.

Classification and OOD Detection We demonstrate the ability of GWI to perform image classifications on Fashion MNIST [Xiao et al., 2017] and CIFAR-10 [Krizhevsky et al., 2009]. We compare

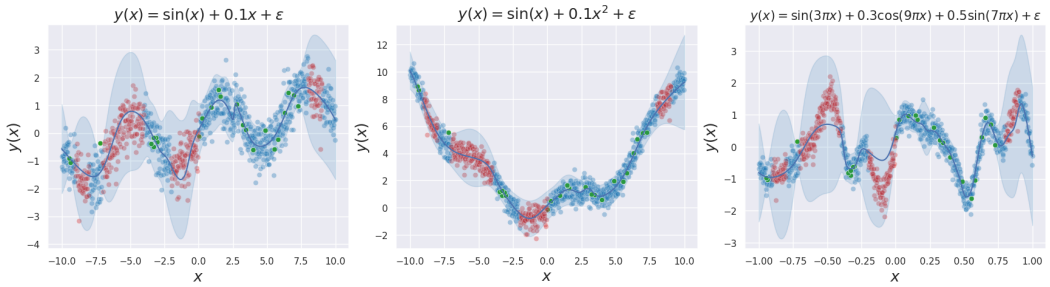


Figure 1: ■ : Training data ■ : Unseen data ■ : Inducing points

We query the above functions at $N = 1000$ equidistant points and add white noise with $\epsilon \sim \mathcal{N}(0, 0.5^2)$. We use $M = 30$ inducing points and train our method as described in Appendix A.7. The plot shows $m_Q(x) \pm 1.96\sqrt{\mathbb{V}[Y^*(x)|Y]}$ where $\mathbb{V}[Y^*(x)|Y]$ is the posterior predictive variance given as $r(x, x) + \sigma^2$.

Dataset	N	D	GWI SVGP	GWI DNN-SVGP	FVI	VIP-BNN	VIP-NP	BBB	VDO	$\alpha = 0.5$	FBNN	**EXACT GP**
BOSTON	506	13	2.8±0.31	**2.27±0.06**	2.33±0.04	2.45±0.04	2.45±0.03	2.76±0.04	2.63±0.10	2.45±0.02	2.30±0.10	2.46±0.04
CONCRETE	1030	8	3.24±0.09	**2.64±0.06**	2.88±0.06	3.02±0.02	3.13±0.02	3.28±0.01	3.23±0.01	3.06±0.03	3.09±0.01	3.05±0.02
ENERGY	768	8	1.81±0.19	0.91±0.12	0.58±0.05	**0.56±0.04**	0.60±0.03	2.17±0.02	1.13±0.02	0.95±0.09	0.68±0.02	0.54±0.02
KIN8NM	8192	8	-0.86±0.38	**-1.2±0.03**	-1.15±0.01	-1.12±0.01	-1.05±0.00	-0.81±0.01	-0.83±0.01	-0.92±0.02	N/A±0.00	N/A±0.00
POWER	9568	4	3.35±0.22	2.74±0.02	**2.69±0.00**	2.92±0.00	2.90±0.00	2.83±0.01	2.88±0.00	2.81±0.00	N/A±0.00	N/A±0.00
PROTEIN	45730	9	**2.84±0.04**	2.87±0.0	2.85±0.00	2.87±0.00	2.96±0.02	3.00±0.00	2.99±0.00	2.90±0.00	N/A±0.00	N/A±0.00
RED WINE	1588	11	0.97±0.02	**0.76±0.08**	0.97±0.06	0.97±0.02	1.20±0.04	1.01±0.02	0.97±0.02	1.01±0.02	1.04±0.01	0.26±0.03
YACHT	308	6	2.37±0.55	0.29±0.1	0.59±0.11	**-0.02±0.07**	0.59±0.13	1.11±0.04	1.22±0.18	0.79±0.11	1.03±0.03	0.10±0.05
NAVAL	11934	16	**-7.25±0.08**	-6.76±0.1	-7.21±0.06	-5.62±0.04	-4.11±0.00	-2.80±0.00	-2.80±0.00	-2.97±0.14	-7.13±0.02	N/A±0.00
Mean Rank			5.5	**2.06**	2.22	3.33	4.94	7	6.11	4.83		

Table 1: The table shows the average test NLL on several UCI regression datasets. We train on random 90% of the data and predict on 10%. This is repeated 10 times and we report mean and standard deviation. The results for our competitors are taken from Ma and Hernández-Lobato [2021].

to FVI, mean-field variational inference (MVFI) [Blundell et al., 2015], maximum a posteriori approximation (MAP), K-FAC Laplace-GNN [Martens and Grosse, 2015] and its dampened version [Ritter et al., 2018].

We also assess the ability of our model to perform out-of-distribution detection using in-distribution (ID) / out of-distribution (OOD) pairs given as FashionMNIST/MNIST and CIFAR10/SVNH. Following the setting of Osawa et al. [2019], Immer et al. [2021] we calculate the area under the curve (AUC) of a binary out-of-distribution classifier based on predictive entropies. Results are shown in Table 2.

	FMNIST			CIFAR 10		
Model	Accuracy	NLL	OOD-AUC	Accuracy	NLL	OOD-AUC
GWI-net	**93.25 ±0.09**	**0.250 ±0.00**	**0.959 ±0.01**	**83.82 ±0.00**	**0.553 ±0.00**	0.618 ±0.00
FVI	91.60±0.14	0.254±0.05	0.956±0.06	77.69 ±0.64	0.675±0.03	0.883±0.04
MFVI	91.20±0.10	0.343±0.01	0.782±0.02	76.40±0.52	1.372±0.02	0.589±0.01
MAP	91.39±0.11	0.258±0.00	0.864±0.00	77.41±0.06	0.690±0.00	0.809±0.01
KFAC-LAPLACE	84.42±0.12	0.942±0.01	0.945±0.00	72.49±0.20	1.274±0.01	0.548±0.01
RITTER et al.	91.20±0.07	0.265±0.00	0.947±0.00	77.38±0.06	0.661±0.00	0.796±0.00

Table 2: We report average accuracy, NLL and OOD-AUC on test data for 10 different train/test splits. The results for FVI are obtained from Ma and Hernández-Lobato [2021] and for MAP, KFAC and Ritter et al. results are provided in Immer et al. [2021] .

Our method performs best in all categories on the Fashion MNIST dataset achieving state-of-the-art results. On CIFAR10 we obtain the highest accuracy and best NLL by a significant margin and perform competitively in the OOD detection task.

6 Conclusion

In this paper, we developed a framework for generalized variational inference in infinite-dimensional function spaces. We leveraged the function space perspective to develop a new inference approach combining Gaussian measures and Wasserstein distance with predictive performance of deep neural networks, yielding principled uncertainty quantification. The value of our method was demonstrated on several benchmark datasets.

References

B. Adlam, J. Snoek, and S. L. Smith. Cold posteriors and aleatoric uncertainty. *arXiv preprint arXiv:2008.00029*, 2020.

L. Ambrosio, N. Gigli, and G. Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2005.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

P. Billingsley. *Probability and measure*. John Wiley & Sons, 2008.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

V. Bogachev. *Gaussian Measures*. American Mathematical Society, 1998.

C. Brislawn. Traceable integral kernels on countably generated measure spaces. *Pacific Journal of Mathematics*, 150(2):229–240, 1991.

D. R. Burt, S. W. Ober, A. Garriga-Alonso, and M. van der Wilk. Understanding variational inference in function-space. *arXiv preprint arXiv:2011.09421*, 2020.

T. Chen, E. Fox, and C. Guestrin. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691. PMLR, 2014.

C.-A. Cheng and B. Boots. Variational inference for gaussian process models with linear complexity. *Advances in Neural Information Processing Systems*, 30, 2017.

G. Da Prato and J. Zabczyk. *Stochastic equations in infinite dimensions*. Cambridge university press, 2014.

N. Durrande, J. Hensman, M. Rattray, and N. D. Lawrence. Detecting periodicities with gaussian processes. *PeerJ Computer Science*, 2:e50, 2016.

V. Dutordoir, N. Durrande, and J. Hensman. Sparse gaussian processes with spherical harmonic features. In *International Conference on Machine Learning*, pages 2793–2802. PMLR, 2020.

D. Duvenaud. The kernel cookbook: Advice on covariance functions. *URL https://www. cs. toronto. edu/duvenaud/cookbook*, 2014.

A. Foong, D. Burt, Y. Li, and R. Turner. On the expressiveness of approximate inference in bayesian neural networks. *Advances in Neural Information Processing Systems*, 33:15897–15908, 2020.

V. Fortuin, A. Garriga-Alonso, F. Wenzel, G. Rätsch, R. Turner, M. van der Wilk, and L. Aitchison. Bayesian neural network priors revisited. *arXiv preprint arXiv:2102.06571*, 2021.

T. Fushiki. Bootstrap prediction and bayesian prediction under misspecified models. *Bernoulli*, 11 (4):747–758, 2005.

Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

J. Gardner, G. Pleiss, K. Q. Weinberger, D. Bindel, and A. G. Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in neural information processing systems*, 31, 2018.

M. Gelbrich. On a formula for the l2 wasserstein metric between measures on euclidean and hilbert spaces. *Mathematische Nachrichten*, 147(1):185–203, 1990.

S. Ghosal and A. Van der Vaart. *Fundamentals of nonparametric Bayesian inference*, volume 44. Cambridge University Press, 2017.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

J. Hensman, N. Durrande, A. Solin, et al. Variational fourier features for gaussian processes. *J. Mach. Learn. Res.*, 18(1):5537–5588, 2017.

G. E. Hinton and D. Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13, 1993.

M. Hladnik and M. Omladič. Spectrum of the product of operators. *Proceedings of the American Mathematical Society*, 102(2):300–302, 1988.

B. R. Hunt, T. Sauer, and J. A. Yorke. Prevalence: a translation-invariant "almost every" on infinite-dimensional spaces. *Bulletin of the American mathematical society*, 27(2):217–238, 1992.

A. Immer, M. Korzepa, and M. Bauer. Improving predictions of bayesian neural nets via local linearization. In *International Conference on Artificial Intelligence and Statistics*, pages 703–711. PMLR, 2021.

C. Jidling, N. Wahlström, A. Wills, and T. B. Schön. Linearly constrained gaussian processes. *Advances in Neural Information Processing Systems*, 30, 2017.

L. V. Kantorovich. Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422, 1960.

A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

M. Khan, D. Nielsen, V. Tangkaratt, W. Lin, Y. Gal, and A. Srivastava. Fast and scalable bayesian deep learning by weight-perturbation in adam. In *International Conference on Machine Learning*, pages 2611–2620. PMLR, 2018.

J. Knoblauch, J. Jewson, and T. Damoulas. Generalized variational inference: Three arguments for deriving new posteriors. *arXiv preprint arXiv:1904.02063*, 2019.

F. Komaki. On asymptotic properties of predictive distributions. *Biometrika*, 83(2):299–313, 1996.

A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

A. Kukush. *Gaussian measures in Hilbert space: construction and properties*. John Wiley & Sons, 2020.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Y. Li and Y. Gal. Dropout inference in bayesian neural networks with alpha-divergences. In *International conference on machine learning*, pages 2052–2061. PMLR, 2017.

Y. Li and R. E. Turner. Gradient estimators for implicit models. *arXiv preprint arXiv:1705.07107*, 2017.

M. Lifshits. Lectures on gaussian processes. In *Lectures on Gaussian Processes*, pages 1–117. Springer, 2012.

C. Ma and J. M. Hernández-Lobato. Functional variational inference based on stochastic process generators. *Advances in Neural Information Processing Systems*, 34, 2021.

C. Ma, Y. Li, and J. M. Hernández-Lobato. Variational implicit processes. In *International Conference on Machine Learning*, pages 4222–4233. PMLR, 2019.

W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson. A simple baseline for bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

J. Martens and R. Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pages 2408–2417. PMLR, 2015.

A. G. d. G. Matthews. *Scalable Gaussian process inference using variational methods*. PhD thesis, University of Cambridge, 2017.

A. G. d. G. Matthews, J. Hensman, R. Turner, and Z. Ghahramani. On sparse variational methods and the kullback-leibler divergence between stochastic processes. In *Artificial Intelligence and Statistics*, pages 231–239. PMLR, 2016.

A. G. d. G. Matthews, M. Rowland, J. Hron, R. E. Turner, and Z. Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

R. M. Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

K. Osawa, S. Swaroop, M. E. E. Khan, A. Jain, R. Eschenhagen, R. E. Turner, and R. Yokota. Practical deep learning with bayesian principles. *Advances in neural information processing systems*, 32, 2019.

V. M. Panaretos and Y. Zemel. Statistical aspects of wasserstein distances. *Annual review of statistics and its application*, 6:405–431, 2019.

R. Ramamoorthi, K. Sriram, and R. Martin. On posterior concentration in misspecified models. *Bayesian Analysis*, 10(4):759–789, 2015.

C. E. Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

H. Ritter, A. Botev, and D. Barber. A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, volume 6. International Conference on Representation Learning, 2018.

T. G. Rudner, Z. Chen, and Y. Gal. Rethinking function-space variational inference in bayesian neural networks. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2020.

H. Salimbeni, C.-A. Cheng, B. Boots, and M. Deisenroth. Orthogonally decoupled variational gaussian processes. *Advances in neural information processing systems*, 31, 2018.

F. Schneider, L. Balles, and P. Hennig. Deepobs: A deep learning optimizer benchmark suite. *arXiv preprint arXiv:1903.05499*, 2019.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958, 2014.

S. Sun, G. Zhang, J. Shi, and R. Grosse. Functional variational bayesian neural networks. *arXiv preprint arXiv:1903.05779*, 2019.

M. Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial intelligence and statistics*, pages 567–574. PMLR, 2009.

B.-H. Tran, S. Rossi, D. Milios, and M. Filippone. All you need is a good functional prior for bayesian deep learning. *arXiv preprint arXiv:2011.12829*, 2020.

M. Van der Wilk, C. E. Rasmussen, and J. Hensman. Convolutional gaussian processes. *Advances in Neural Information Processing Systems*, 30, 2017.

M. van der Wilk, M. Bauer, S. John, and J. Hensman. Learning invariances using the marginal likelihood. *Advances in Neural Information Processing Systems*, 31, 2018.

K. Wang, G. Pleiss, J. Gardner, S. Tyree, K. Q. Weinberger, and A. G. Wilson. Exact gaussian processes on a million data points. *Advances in Neural Information Processing Systems*, 32, 2019.

M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

F. Wenzel, K. Roth, B. S. Veeling, J. Świątkowski, L. Tran, S. Mandt, J. Snoek, T. Salimans, R. Jenatton, and S. Nowozin. How good is the bayes posterior in deep neural networks really? *arXiv preprint arXiv:2002.02405*, 2020.

V. Wild and G. Wynne. Variational gaussian processes: A functional analysis view. *arXiv preprint arXiv:2110.12798*, 2021.

A. G. Wilson and P. Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708, 2020.

A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378. PMLR, 2016.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Appendix A.10
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Nothing to discuss
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix A.1-A.6
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See footnote in introduction
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix A.7 and A.8
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix A.11
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

A.1 Bayesian Inference as an Optimization Problem for an Infinite-Dimensional Prior Measure

Let E be a (infinite dimensional) Polish space and $\mathcal{B}(E)$ the Borel σ-algebra on E. We denote the set of Borel probability measures on $\mathcal{B}(E)$ as $\mathcal{P}(E)$ and choose a fixed prior measure $P \in \mathcal{P}(E)$. The likelihood is described by a Markov kernel function $p : \mathcal{Y} \times E \to [0, \infty)$ with

$$(y, f) \mapsto p(y|f), \tag{19}$$

where $\mathcal{Y} \subset \mathbb{R}^N$ is Borel measurable. The prior and the likelihood induce for any fixed $y \in \mathcal{Y}$ a posterior measure denoted as $\widehat{P} \in \mathcal{P}(E)$ [Ghosal and Van der Vaart, 2017, Chapter 1.3].

The next theorem shows that this posterior measure is the solution to a certain optimization problem.

Theorem 2 (Bayes Posterior as optimization)**.** *The Bayesian posterior measure $\widehat{P}$ is given as*

$$\widehat{P} = \underset{Q \in \mathcal{P}(E)}{\textit{argmin}} \left\{ -\mathbb{E}_Q \big[\log p(y|F) \big] + \mathbb{D}_{KL}(Q, P) \right\} \tag{20}$$

for any fixed prior measure $P \in \mathcal{P}(E)$ and fixed $y \in \mathcal{Y}$ such that $f \in E \mapsto p(y|f) > 0$.

Proof. According to Bayes rule in infinite dimensions [Ghosal and Van der Vaart, 2017, Chapter 1.3] we know that $\widehat{P}$ is dominated by P with Radon-Nikodym derivative given as

$$\frac{d\widehat{P}}{dP}(f) = \frac{p(y|f)}{p(y)}, \tag{21}$$

for $f \in E$ where $p(y) := \int p(y|F = f)\, dP(f)$ is the marginal likelihood for y. The reverse is also true and P is dominated by $\widehat{P}$. We prove this by contraposition and therefore assume that $P(A) > 0$ for some $A \in \mathcal{B}(E)$. From Bayes rule we know that

$$\widehat{P}(A) = \int_A \frac{p(y|f)}{p(y)}\, dP(f) > 0 \tag{22}$$

as the integrand is positive by assumption and $P(A) > 0$. This gives $\widehat{P}(A) > 0$ and therefore that P is dominated by $\widehat{P}$. In this case standard rules for Radon-Nikodym derivatives give that

$$\frac{dP}{d\widehat{P}}(f) = \frac{p(y)}{p(y|f)}, \tag{23}$$

for $f \in E$. Note that without loss of generality we can assume that the optimal $Q \in \mathcal{P}(E)$ is dominated by P (and therefore also dominated by $\widehat{P}$) since otherwise (20) is infinite by definition of the KL divergence. For such a Q dominated by P it holds that

$$\begin{aligned}
L(Q) &:= -\mathbb{E}_Q \big[\log p(y|F) \big] + \mathbb{D}_{KL}(Q, P) && (24)\\
&= -\int \log p(y|f)\, dQ(f) + \int \log \frac{dQ}{dP}(f)\, dQ(f) && (25)\\
&= -\int \log p(y|f)\, dQ(f) + \int \log \frac{dQ}{d\widehat{P}}(f)\, dQ(f) + \int \log \frac{d\widehat{P}}{dP}(f)\, dQ(f), && (26)
\end{aligned}$$

where the last line follows from the chain rule for Radon-Nikodym derivatives. We further have

$$\begin{aligned}
L(Q) &= -\int p(y|f)\, dQ(f) + \mathbb{D}_{KL}(Q, \widehat{P}) + \int \frac{p(y|f)}{p(y)}\, dQ(f) \qquad \text{(Bayes Rule)} && (27)\\
&= \mathbb{D}_{KL}(Q, \widehat{P}) + p(y) && (28)\\
&\geq p(y), && (29)
\end{aligned}$$

since $\mathbb{D}_{KL}(Q, P) \geq 0$, with equality if and only if $Q = \widehat{P}$. This proves the claim. □

A.2 Pointwise Evaluation as Weak Limit

To outline the problem briefly: If $F \sim \mathcal{N}(m, C)$ is a GRE with mean $m \in L^2(\mathcal{X}, \rho, \mathbb{R})$ and covariance operator C as defined in (9) then it is in general unclear what the distribution of $F(x)$ would be for a fixed $x \in \mathcal{X}$. The technical reason is that the pointwise evaluation $\pi_x : L^2(\mathcal{X}, \rho, \mathbb{R}) \to \mathbb{R}$, i.e.

$$\pi_x(f) := f(x) \tag{30}$$

is not well-defined. An element g of the space $L^2(\mathcal{X}, \rho, \mathbb{R})$ is an equivalence class and only identifiable up to a ρ-nullset. This means that the definition of π_x in (30) makes no sense whenever $\rho(\{x\}) = 0$ which is the case whenever ρ has a pdf w.r.t. the Lebesgue measure.

However, we will remedy this situation by defining for a fixed $x \in \mathcal{X}$

$$F(x) := \lim_{n\to\infty} \langle F, h_{n,x}\rangle_2 \tag{31}$$

where $h_{n,x} \in L^2(\mathcal{X}, \rho, \mathbb{R})$ is an appropriately chosen sequence and the limit is to be understood as convergence in distribution of the sequence of scalar random variables $\langle F, h_{n,x}\rangle_2$.

Theorem 3. *Let $F \sim \mathcal{N}(m, C)$ be a GRE in $\mathcal{L}^2(\mathcal{X}, \rho, \mathbb{R})$ with mean $m \in L^2(\mathcal{X}, \rho, \mathbb{R})$ and covariance operator C as defined in* (9)*. Assume that ρ is a probability measure on $\mathcal{X} \subset \mathbb{R}^D$ and that ρ is absolutely continuous with respect to the Lebesgue measure λ on $\mathbb{R}^D$ with pdf ρ'. Denote the support of the measure ρ by $supp(\rho)$ and assume that x is an arbitrary point in the interior of $supp(\rho)$ such that m, k and ρ' are continuous at x.*

Let

$$\eta(t) = \begin{cases} \exp\Big(-\frac{1}{1-|t|^2}\Big) & \text{if } |t| < 1, \\ 0 & \text{if } |t| \geq 1. \end{cases} \tag{32}$$

be the so called standard molifier and note that η is smooth with $\int \eta(t)\,dt = 1$. We further define the sequence $h_{n,x}(t) := \eta\big(n(t-x)\big)/\rho'(t)$ for $n \in \mathbb{N}$, $t \in supp(\rho)$ and $h_{n,x} = 0$ for $t \notin supp(\rho)$. Then

$$\langle F, h_{n,x}\rangle_2 \xrightarrow{\mathcal{D}} \mathcal{N}\big(m(x), k(x,x)\big) \tag{33}$$

for $n \to \infty$ where $\xrightarrow{\mathcal{D}}$ denotes convergence in distribution.

Proof. Note that $supp(h_{n,x}) = B_{1/n}(x) := \{t \in \mathbb{R}^D : |t - x| \leq \frac{1}{n}\}$ and $B_{1/n}(x) \subset supp(\rho)$ for large enough $n \in \mathbb{N}$ since x is from the interior of $supp(\rho)$. This means that $h_{n,x} \in L^2(\mathcal{X}, \rho, \mathbb{R})$ for large enough n as

$$\int h_{n,x}(t)\,d\rho(t) = \int_{supp(\rho)} \left(\frac{\eta\big(n(t-x)\big)}{\rho'(t)}\right)^2 \rho'(t)\,d\lambda(t) \tag{34}$$

$$= \int_{supp(\rho)} \frac{\eta\Big(n(t-x)\Big)}{\rho'(t)}\,dt \tag{35}$$

$$= \int_{B_{1/n}(x)} \frac{\eta\Big(n(t-x)\Big)}{\rho'(t)}\,dt. \tag{36}$$

The last expression is finite for large enough n because the integrand is continuous at x. According to the definition of of GREs we therefore conclude that

$$\langle F, h_{n,x}\rangle_2 \sim \mathcal{N}\big(\langle m, h_{n,x}\rangle_2, \langle Ch_{n,x}, h_{n,x}\rangle_2\big) \tag{37}$$

for large enough $n \in \mathbb{N}$.

The next statement we show is that $m_n(x) := \langle m, h_{n,x}\rangle_2 \to m(x)$ for $n \to \infty$. To this end notice that

$$|m_n(x) - m(x)| = |\int_{B_{1/n}(x)} h_{n,x}(t)\big(m(x) - m(t)\big)\,d\rho(t)| \tag{38}$$

$$\leq \int_{B_{1/n}(x)} \eta\Big(n(t-x)\Big)|m(x)-m(t)|\,dt. \tag{39}$$

Let now $\epsilon > 0$ be arbitrary. For n large enough we $|m(x)-m(t)| \leq \epsilon$ for all $t \in B_{1/n}(x)$ due to the continuity of m in x. This immediately implies

$$\int_{B_{1/n}(x)} \eta\Big(n(t-x)\Big)|m(x)-m(t)|\,dt \leq \epsilon \int_{B_{1/n}(x)} \eta\Big(n(t-x)\Big)\,dt = \epsilon, \tag{40}$$

for large enough n which shows the convergence of $m_n(x)$ to $m(x)$.

A similar argument shows that $k_n(x,x) := \langle Ch_{n,x}, h_{n,x}\rangle_2 \to k(x,x)$ for $n \to \infty$.

We therefore conclude that

$$\begin{aligned}
\langle F, h_{n,x}\rangle_2 &= \langle F, h_{n,x}\rangle_2 - m_n(x) + m_n(x) && (41)\\
&= \sqrt{k_n(x,x)}\,\underbrace{\frac{\langle F, h_{n,x}\rangle_2 - m_n(x)}{\sqrt{k_n(x,x)}}}_{\sim\mathcal{N}(0,1)} + m_n(x) && (42)\\
&\xrightarrow{\mathcal{D}} \mathcal{N}\big(m(x), k(x,x)\big) && (43)
\end{aligned}$$

for $n \to \infty$ due to Slutsky's theorem. □

According to Theorem 3 we can simply define $F(x) \sim \mathcal{N}(m(x), k(x,x))$ for all x in the interior of the support of ρ if m, k and ρ' are continuous at x. These are mild assumptions and we can typically assume that they are satisfied in practice.

A.3 The Wasserstein Metric for Probability Measures

Let E be a Polish space. For $p \geq 1$, let $P_p(E)$ denote the collection of all probability measures μ on E with finite p^{th} moment, that is, there exists some x_0 in M such that:

$$\int_M d(x,x_0)^p \,\mathrm{d}\mu(x) < \infty. \tag{44}$$

The pth Wasserstein distance between two probability measures μ and ν in $P_p(E)$ is defined as

$$W_p(\mu,\nu) := \left(\inf_{\gamma\in\Gamma(\mu,\nu)} \int_{E\times E} d(x,y)^p \,\mathrm{d}\gamma(x,y)\right)^{1/p}, \tag{45}$$

where $\Gamma(\mu,\nu)$ denotes the collection of all measures on $E \times E$ with marginals μ and ν on the first and second arguments respectively.

More details about the Wasserstein distance can be found in Chapter 7 of Ambrosio et al. [2005].

A.4 A Tractable Approximation of the Wasserstein Metric

Recall that the Wasserstein metric for the two Gaussian measures $P = \mathcal{N}(m_P, C_P)$ and $Q = \mathcal{N}(m_Q, C_Q)$ on the Hilbert space $H = L^2(\mathcal{X}, \rho, \mathbb{R})$ is given as

$$W_2^2(P,Q) = \|m_P - m_Q\|_2^2 + tr(C_P) + tr(C_Q) - 2\cdot tr\Big[\big(C_P^{1/2} C_Q C_P^{1/2}\big)^{1/2}\Big]. \tag{46}$$

Further the operators C_P and C_Q are defined through trace-class kernels k and r as described in Section 3.1. We will now discuss how to approximate each term in (46).

First, note that

$$\|m_P - m_Q\|_2^2 = \int \big(m_P(x) - m_Q(x)\big)^2\,d\rho(x) \approx \frac{1}{N}\sum_{n=1}^{N} \big(m_P(x_n) - m_Q(x_n)\big)^2, \tag{47}$$

which follows by replacing the true input distribution with the empirical data distribution. Second, note that under very general conditions on k and ρ it holds that [Brislawn, 1991]

$$tr(C_P) = \int k(x,x)\, d\rho(x) \tag{48}$$

and similarly for C_Q. Again by replacing ρ with the empirical data distribution we obtain natural estimators:

$$tr(C_P) \approx \frac{1}{N}\sum_{n=1}^{N} k(x_n, x_n), \tag{49}$$

$$tr(C_Q) \approx \frac{1}{N}\sum_{n=1}^{N} r(x_n, x_n). \tag{50}$$

Denote by $\lambda_n(C)$ the n-th eigenvalue of a positive, self-adjoint operator C. By definition of the trace and the square root of an operator we have

$$tr\Big[\big(C_P^{1/2} C_Q C_P^{1/2}\big)^{1/2}\Big] = \sum_{n=1}^{\infty} \sqrt{\lambda_n\Big(C_P^{1/2} C_Q C_P^{1/2}\Big)} \tag{51}$$

$$= \sum_{n=1}^{\infty} \sqrt{\lambda_n\Big(C_Q C_P\Big)}, \tag{52}$$

where the second line follows from the fact that the operator $C_Q C_P$ has the same eigenvalues as $C_P^{1/2} C_Q C_P^{1/2}$ [Hladnik and Omladič, 1988, Proposition 1]. The operator $C_Q C_P$ is given as

$$C_Q C_P g(x) = \int r(x,x')(C_P f)(x')\, d\rho(x') \tag{53}$$

$$= \int r(x,x')\Big(\int k(x',t) f(t) d\rho(t)\Big)\, d\rho(x') \tag{54}$$

$$= \int\int r(x,x') k(x',t) f(t)\, d\rho(x') d\rho(t) \tag{55}$$

$$= \int (r * k)(x,t) f(t)\, d\rho(t), \tag{56}$$

where we define

$$(r * k)(x,t) := \int r(x,x') k(x',t)\, d\rho(x') \tag{57}$$

for all $x, t \in \mathcal{X}$. This means that $C_Q C_P$ is also an integral operator with (non-symmetric) kernel $r * k$. We again replace ρ with $\widehat{\rho}$ to obtain

$$\widehat{(r * k)}(x,t) = \frac{1}{N}\sum_{n=1}^{N} r(x, x_n) k(x_n, t). \tag{58}$$

The spectrum of $C_Q C_P$ can now be approximated by the spectrum of the matrix $\frac{1}{N}\widehat{(r * k)}(X,X)$ [Rasmussen, 2003, cf. Chapter 4.3.2] or $\frac{1}{N_S}\widehat{(r * k)}(X_S, X_S)$ where X_S is a subsample of the data points X of size $N_S < N$. If we plug this approximation into (52) we obtain

$$tr\Big[\big(C_P^{1/2} C_Q C_P^{1/2}\big)^{1/2}\Big] \approx \sum_{m=1}^{N_S} \sqrt{\lambda_m\big(\frac{1}{N_S}\widehat{(r * k)}(X_S, X_S)\big)} \tag{59}$$

$$= \frac{1}{\sqrt{N_S}} \sum_{m=1}^{N_S} \sqrt{\lambda_m\Big(\frac{1}{N} r(X_S, X) k(X, X_S)\Big)}, \tag{60}$$

which is the last expression that we had to approximate.

Note that since $C_Q C_P$ has the same spectrum as the self-adjoint, positive, trace-class operator $C_P^{1/2} C_Q C_P^{1/2}$ we know that its eigenvalues are real, positive and converge to zero.

A.5 Generalized Loss for Regression in Batch Mode

The batch version of the generalized loss is given as:

$$\widehat{\mathcal{L}} = \frac{N}{2}\log(2\pi\sigma^2) + \frac{N}{N_B}\sum_{b=1}^{N_B}\frac{\big(y_{n_b} - m_Q(x_{n_b})\big)^2 + r(x_{n_b}, x_{n_b})}{2\sigma^2} + \frac{1}{N_B}\sum_{b=1}^{N_B}\big(m_P(x_{n_b}) - m_Q(x_{n_b})\big)^2 \tag{61}$$

$$+ \frac{1}{N_B}\sum_{b=1}^{N_B} k(x_{n_b}, x_{n_b}) + \frac{1}{N_B}\sum_{b=1}^{N_B} r(x_{n_b}, x_{n_b}) - \frac{2}{\sqrt{N_B N_S}}\sum_{s=1}^{N_S}\sqrt{\lambda_s\big(r(X_S, X_B)k(X_B, X_S)\big)}, \tag{62}$$

$N_B \in \mathbb{N}$ is the batch-size. The indices $n_1, \ldots, n_{N_B}$ are the batch-indices and X_B is the batch matrix.

A.6 GWI for (Multiclass) Classification

Let $\{(x_n, y_n)\}_{n=1}^N \subset \mathcal{X} \times \mathcal{Y}$ be data with $\mathcal{X} \subset \mathbb{R}^D$ and $\mathcal{Y} = \{1, \ldots, J\}$, where $J \in \mathbb{N}$ represents $J \geq 2$ distinct classes.

Model We use the same likelihood for $y := (y_1, \ldots, y_N)$ as described in Chapter 4 of Matthews [2017] which is:

$$p(y|f_1, \ldots, f_J) = \prod_{n=1}^N p(y_n|f_1, \ldots, f_J) \tag{63}$$

with

$$p(y_n|f_1, \ldots, f_J) := h_{y_n}^{\epsilon}\big(f_1(x_n), \ldots, f_J(x_n)\big), \tag{64}$$

for $y_n \in \{1, \ldots, J\}$. The function h_ℓ^ϵ is defined as

$$h_\ell^\epsilon(t_1, \ldots, t_J) \begin{cases} 1 - \epsilon & \text{if } \ell = \operatorname*{argmax}_{j=1,\ldots,J}\{t_j\}, \\ \frac{\epsilon}{J-1} & \text{if otherwise.} \end{cases} \tag{65}$$

for $\ell = 1, \ldots, J$ for $\epsilon > 0$. We chose $\epsilon = 1\%$ in our implementation.

We assume that $F_1, \ldots F_J$ are independent GREs on $L^2(\mathcal{X}, \rho, \mathbb{R})$ with prior means $m_{P,j}$ and prior covariance operators $C_{P,j}$, $j = 1, \ldots, J$.

The variational measures for $F_1, \ldots, F_J$ are assumed to be independent and given as $Q_j = \mathcal{N}\big(m_{Q,j}, C_{Q,j}\big)$ for $j = 1, \ldots, J$. We further write $\mathbb{Q}\Big(\big(F_1(x), \ldots, F_J(x)\big) \in A\Big)$, $A \subset \mathbb{R}^J$ for the variational (posterior) approximation of the probability of the event $\{\big(F_1(x), \ldots, F_J(x)\big) \in A\}$.

This leads to the following expected log-likelihood

$$\mathbb{E}_{\mathbb{Q}}\big[\log p(y|F_1, \ldots, F_J)\big] \tag{66}$$

$$= \sum_{n=1}^N \mathbb{E}_{\mathbb{Q}}\big[\log p(y_n|F_1, \ldots, F_J)\big] \tag{67}$$

$$= \sum_{n=1}^N \log(1-\epsilon)\mathbb{Q}\big(\operatorname*{argmax}_{j=1,\ldots,J}\{F_j(x_n)\} = y_n\big) + \log(\frac{\epsilon}{J-1})\mathbb{Q}\big(\operatorname*{argmax}_{j=1,\ldots,J}\{F_j(x_n)\} \neq y_n\big) \tag{68}$$

$$\approx \sum_{n=1}^N \log(1-\epsilon)S(x_n, y_n) + \log(\frac{\epsilon}{J-1})\big(1 - S(x_n, y_n)\big), \tag{69}$$

with

$$S(x, j) := \frac{1}{\sqrt{\pi}}\sum_{i=1}^I w_i \prod_{l \neq j}\phi\Big(\frac{\sqrt{2r_j(x,x)}\xi_i + m_{Q,j}(x) - m_{Q,l}(x)}{\sqrt{r_l(x,x)}}\Big) \tag{70}$$

for any $x \in \mathcal{X}$, $j = 1, \dots, J$ where $(w_i, \xi_i)_{i=1}^I$ are the weights and roots of the Hermite polynomial of order $I \in \mathbb{N}$. This is the same Gauss-Hermite approximation as described in Chapter 4 of Matthews [2017].

The final objective for multiclass classification is given as

$$\mathcal{L} = -\mathbb{E}_Q\big[\log p(y|F_1, \dots, F_J)\big] + \sum_{j=1}^{J} W_2^2(P_j, Q_j), \tag{71}$$

where the expected log-likelihood is approximated by (69) and each Wasserstein distance $W_2^2(P_j, Q_j)$ can be estimated as in (14)-(15).

Prediction The probability that an unseen point $x^* \in \mathcal{X}$ belongs to class $j \in \{1, \dots, J\}$ is given as

$$\mathbb{Q}(Y^* = j) = (1-\epsilon)S(x^*, j) + \frac{\epsilon}{J-1}\big(1 - S(x^*, j)\big) \tag{72}$$

for any $x^* \in \mathcal{X}$. We predict the class label as maximiser of this probability. If we apply tempering, we simply replace every $r_j(x, x)$ with $T \cdot r_j(x, x)$ for $j = 1, \dots, J$ in the definition of $S(x, j)$.

Negative Log Likelihood The variational approximation to the negative log-likelihood is

$$NLL = -\log\Big[(1-\epsilon)S(x^*, y^*) + \frac{\epsilon}{J-1}\big(1 - S(x^*, y^*)\big)\Big] \tag{73}$$

for any point $x^* \in \mathcal{X}$ for which we know that the class label is $y^* \in \{1, \dots, J\}$.

A.7 Implementation Details: Regression

The Regression model is given as $F \sim \mathcal{N}(0, C)$ and

$$Y_n = F(x_n) + \epsilon_n \tag{74}$$

with $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$, $n = 1, \dots, N$. The covariance operator C_P depends on the choice of a kernel k, i.e. $C_P = C_{P,k}$ for which we use the ARD kernel k given as

$$k(x, x') = \sigma_f^2 \exp\Big(-\frac{1}{2}\sum_{d=1}^{D}\frac{(x_d - x'_d)^2}{\alpha_d^2}\Big) \tag{75}$$

for $x, x' \in \mathbb{R}^D$. We refer to $\sigma_f > 0$ as *kernel scaling factor*, to $\alpha_d > 0$ as *length-scale* for dimension d and to $\sigma > 0$ as *observation noise*.

The data is first randomly split into three categories: training set 80%, validation set 10% and test set 10%. The observations Y are then standardised by subtracting the empirical mean (of the training data) and dividing by the empirical standard deviation (of the training data). The inputs data X is left unaltered.

The number of inducing points The number of inducing points M is treated as a hyperparameter, this means we train the model for each $M \in \{0.5\sqrt{N}, \sqrt{N}, 1.5\sqrt{N}, 2\sqrt{N}\}$ and choose the best model. For GWI: SVGP we use $M \in \{1\sqrt{N}, 2\sqrt{N}, \dots 5\sqrt{N}\}$.

The choice of inducing points The input points $Z_1, \dots, Z_M$ in (18) are sampled independently from the training data X and then fixed for GWI-net. For GWI: SVGP they are only initialised this way and then learned by maximising the generalized loss.

Prior hyperparameters The prior hyperparameters σ_f, $\alpha := (\alpha_1, \dots, \alpha_D)$ and σ are chosen by maximising the marginal log-likelihood for the data $X = Z$ and the corresponding observations, which we denote Y_Z. Note that the marginal log-likelihood is tractable and given as

$$\log p(y_Z) = -\frac{1}{2}\log\Big(\det\big(k(Z,Z) + \sigma^2 I_M\big)\Big) - \frac{1}{2}{y_Z}^T\big(k(Z,Z) + \sigma^2 I_M\big)^{-1} y_Z. \tag{76}$$

and can therefore be evaluated in $\mathcal{O}(M^3) = \mathcal{O}(N\sqrt{N})$. **Variational mean** For GWI-net we use a neural network with $L = 2$ hidden layers, width $D_1 = D_2 = 10$ and tanh as activation function. This follows the set-up of Ma and Hernández-Lobato [2021].

Variational kernel The kernel r which is chosen as described in (18) and therefore depends on the covariance matrix $\Sigma \in \mathbb{R}^{M\times M}$ and the $M \in \mathbb{N}$ inducing points $Z = (Z_1, \dots, Z_M) \in \mathbb{R}^{D\times M}$. We parametrise Σ as $\Sigma = LL^T$ with initialisation

$$L = \text{Chol}\Big(\big(k(Z,Z) + \frac{1}{\sigma^2}k(Z,X)k(X,Z)\big)^{-1}\Big), \tag{77}$$

where $k(Z,X)k(X,Z)$ is approximated by batch-sizing as $\frac{N}{N_B}k(Z,X_B)k(X_B,Z)$. This corresponds to an approximation of the optimal choice for Σ in SVGP [Titsias, 2009].

Parameters in the generalized loss The generalized loss in Appendix A.5 depends further on N_S, N_B and X_S. The batch-size N_B is chosen to be $N_B = 1000$ for $N > 1000$. For $N < 1000$ we use the full training data. The comparison points X_S are sampled independently from the training data X in each iteration. We train here for 1000 epochs on the regression task and 100 epochs on the classification task following Ma and Hernández-Lobato [2021].

Tempering the predictive posterior

Wenzel et al. [2020] observe that the performance of many Bayesian neural networks can be improved by *tempering* the predictive posterior. Tempering refers to a shrinking of the predictive posterior variance by a factor of $\alpha_T \in [0,1]$. This effect has also been observed for Gaussian processes in Adlam et al. [2020] where it can be interpreted as elevating problems that occur from prior misspecification. The prior hyperparameters for the ARD kernel k in (16) are selected by maximising the marginal log-likelihood on a subset of the training data. This procedure may lead to prior misspecification, which is why we decided to temper the predictive posterior, which means that we use the predictive distribution

$$Y^*|Y \sim \mathcal{N}\Big(m_Q(x^*), \alpha_T\big(r(x^*,x^*) + \sigma^2\big)\Big) \tag{78}$$

for an unseen data point $x^* \in \mathcal{X}$. The (tempered) NLL for each data point is given as

$$\text{NLL} := -\log p_{\alpha_T}(y^*|y) \tag{79}$$

$$= \frac{1}{2}\log\Big(\alpha_T \cdot (r(x^*,x^*) + \sigma^2)\Big) + \frac{1}{2}\frac{(y-y^*)^2}{\alpha_T \cdot (r(x^*,x^*) + \sigma^2)} + \frac{1}{2}\log(2\pi). \tag{80}$$

The tempering factor α_T is chosen as minimiser of the average NLL on the validation set. The final predictions on the test set are made using this optimal α_T and (78). Note however that for the NLL numbers reported in Table 1 we add $\log(\widehat{\sigma}_{train})$ to (80) where $\widehat{\sigma}_{train}$ is the empirical standard deviation of the training data. This is done for fair comparison as it is how the NLL is calculated in Ma and Hernández-Lobato [2021].

A.8 Implementation Details: Classification

As described in section (A.6) we use the prior mean functions $m_{P,j}$ and kernels k_j for $j = 1, \dots, J$. For our experiments we chose $m_{P,j} = 0$ for $j = 1, \dots, J$ and $k := k_1 = \dots, k_J$ where k is the ARD kernel in (16).

We use a multi-output neural network for the variational means $m_{Q,j}$ and an SVGP kernel for each r_j , $j = 1, \dots, J$.

The number of inducing points The number of inducing points M is treated as a hyperparameter, this means we train the model for each $M \in \{0.5\sqrt{N}, 0.75\sqrt{N}, \sqrt{N}\}$ and choose the best model.

The choice of inducing points The input points $Z_1, \dots, Z_M$ in (18) are sampled independently from the training data X and then fixed for GWI-net.

Prior hyperparameters The prior hyperparameters are initialised as described in A.7, thus maximising the marginal likelihood of a *regression* model, since the marginal likelihood of our classification model is intractable.

Variational mean We use the same CNN architecture as described in Immer et al. [2021], Schneider et al. [2019] for all models.

Variational kernel Each variational kernel r_j uses the same inducing points Z but gets an individual matrix $\Sigma^j \in \mathbb{R}^{M \times M}$ for $j = 1, \dots, J$. They are all initialised as described in A.7.

Parameters in the generalized loss The generalized loss in Appendix A.5 depends on N_S, N_B and X_S. The batch-size N_B is chosen to be $N_B = 1000$ for $N > 1000$. For $N < 1000$ we use the full training data. The comparison points X_S are sampled independently from the training data X in each iteration. We train 100 epochs on the classification task following Ma and Hernández-Lobato [2021].

Tempering the predictive posterior For the same reasons as outlined in Appendix A.7 we temper the predictive posterior. Recall that the NLL for classification is given as

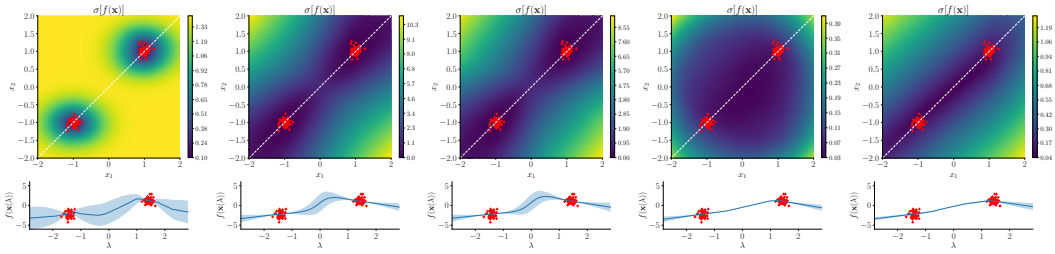
$$NLL = -\log\Big[(1-\epsilon)S(x^*, y^*) + \frac{\epsilon}{J-1}\big(1 - S(x^*, y^*)\big)\Big] \tag{81}$$

for any point $x^* \in \mathcal{X}$ for which we know that the class label is $y^* \in \{1, \dots, J\}$. We use a tempering factor $\alpha_j > 0$ for each variational measure $Q_j \sim \mathcal{N}(m_{Q,j}, \alpha_j r_j)$, $j = 1, \dots, J$. We train the model with $\alpha_j = 1$ for all $j = 1, \dots, J$ and select the tempering factors afterwards as minimiser of the average NLL on the validation set.

A.9 Illustrative Example for Two Dimensional Inputs

In Foong et al. [2020] it is observed that several BNN posterior approximation techniques struggle with the quantification of in-between uncertainty. The red points mark where observations were made and it is clear that mean-field variational inference (MVFI) [Hinton and Van Camp, 1993] and Monte Carlo Dropout (MCDO) [Gal and Ghahramani, 2016] exhibit unjustifiably high posterior certainty in the area where no observations are made. This is a pathology of the approximation technique as the true Bayesian posterior which is approximated to very high precision by Hamiltonian Monte Carlo (HMC) [Neal, 2012] or the infinite-width GP limit [Matthews et al., 2018] do not display such behaviour.

In Figure 2 our method GWI-net is displayed next to the methods described in Foong et al. [2020]. As one can observe our model is keenly aware of its limited ability to predict points in-between the two clusters of observed data points.



(a) GWI (b) Inf-width limit GP (c) HMC (d) MFVI (e) MCDO

Figure 2: Regression on a 2D synthetic dataset (red crosses). The colour plots show the standard deviation of the output, $\sigma[f(\mathbf{x})]$, in 2D input space. The plots beneath show the mean with 2-standard deviation bars along the dashed white line (parameterised by λ). MFVI and MCDO are overconfident for $\lambda \in [-1, 1]$.

A.10 Model Misspecification in Gaussian Wasserstein Inference

The generalized loss in Appendix A.5 is a valid optimization target for any $m_P, m_Q \in L^2(\mathcal{X}, \rho, \mathbb{R})$ and any trace-class kernels k and r. This gives the user a lot of abilities to specify different models, by experimenting with various choices, specifically for m_Q and r. However with great power comes great responsibility: it is quite easy to misspecify GWI. To illustrate the issue let us use a periodic kernel k [Duvenaud, 2014] given as

$$k(x, x') := \sigma_f^2 \exp\big(-\frac{1}{\alpha^2}\sin^2(\pi|x - x'|/p)\big) \tag{82}$$

and the SVGP kernel r in (18). By the definition of r the uncertainty will be low for points *similiar* to the inducing points Z, i.e. for points $x \in \mathcal{X}$ $k(x, z_m) \approx \sigma_f^2$ for all $m = 1, \dots, M$. A problem now occurs, if the posterior mean m_Q does not respect the knowledge embedded in k and r. Lets for example use a simple fully connected deep neural network m_Q and choose the point $x^* := z_1 + 10p$. Assume further that $z_1, \dots, z_M < x^*$. Then we get $k(x^*, z_m) = k(z_1, z_m)$ for all $m = 1, \dots, M$ due to the periodicity of $\sin(x)$ and therefore $r(x^*, x^*) = r(z_1, z_1)$. It is however very unlikely that the neural network will predict $m_Q(z_1)$ as well as $m_Q(x^*)$ since it is unaware of this periodicity.

This small example should illustrate that it is crucial that m_Q is compatible with the prior knowledge reflected in k and r. However, note that this problem is not present for our model, GWI-net. The ARD kernel encodes the inductive bias that the underlying function is infinitely differentiable and that points close to each other have highly correlated functional outputs. A simple fully connected DNN with tanh activation function is indeed smooth and further it is reasonable to assume that predictions are more unreliable the further they are from the data (as measured by the squared euclidean distance). The ARD kernel is in this sense compatible with a fully connected DNN.

It shall be noted that the DNN used for the classification examples in (5) used convolutional layers as explained in Appendix A.8. This can be understood as embedding prior knowledge about translation equivariance into the DNN [Goodfellow et al., 2016, Chapter 9.4]. It might therefore be desirable to use a prior kernel k that embeds similar properties such as the kernel suggested by Van der Wilk et al. [2017]. We considered this to be beyond the scope of this paper but the interaction of DNN architecture and the choice of prior kernels is an interesting avenue for future research.

A.11 Details on computational resources used

For all our experiments, we distributed our jobs across 8 Nvidia V100 cards.

A.12 Additional plots for 1D experiments

We provide additional 1D experiments in Figure 3 with additional comparisons to GWI-KRR and SVGP.

A.13 Empirical estimation error of 2-Wasserstein distance

The approximation quality of the 2-Wasserstein distance is determined by the approximation quality of the spectrum of the appearing covariance operators. For most kernels in practice like SE or Matern kernel, the spectrum decays very quickly, which is why using the first 100 eigenvalues often empirically seems to be sufficient to approximate the spectrum and therefore the 2-Wasserstein distance. We plot the magnitude of the first 100 positive eigenvalues (sorted on magnitude) for datasets BOSTON, CONCRETE, ENERGY, WINE and YACHT in Figure 4.

We see in Figure 4 that eigenvalues indeed decay fast.

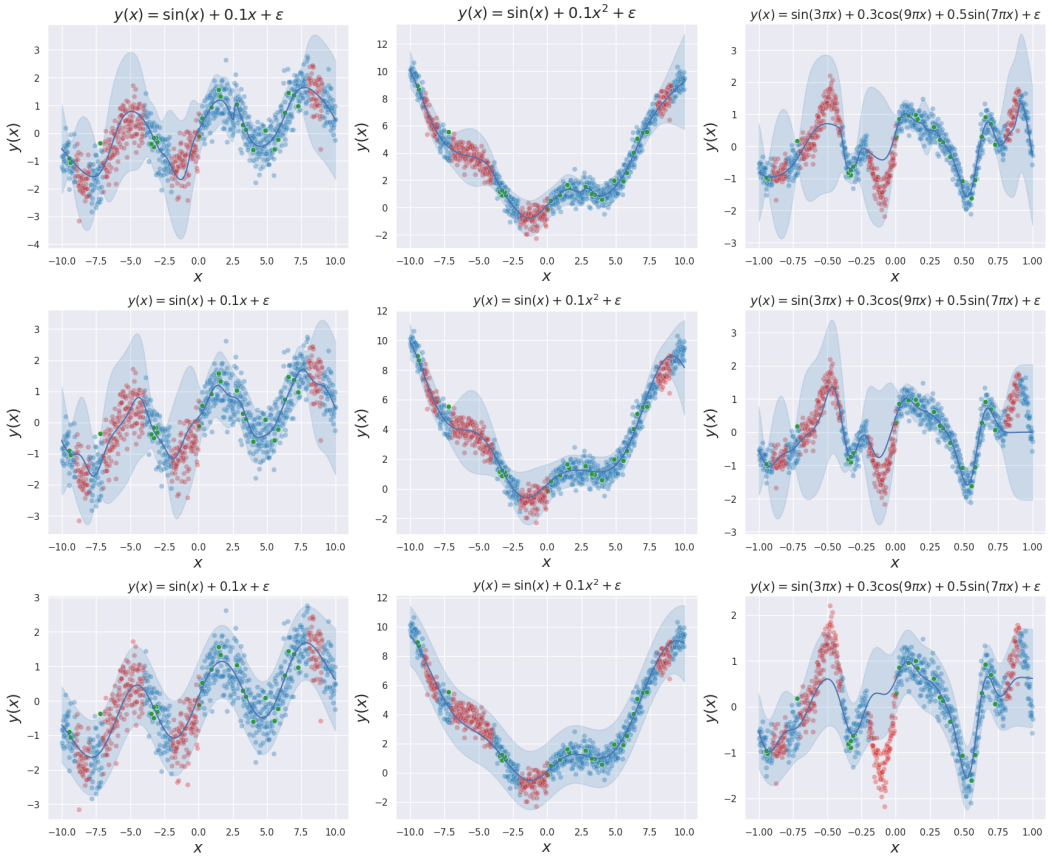
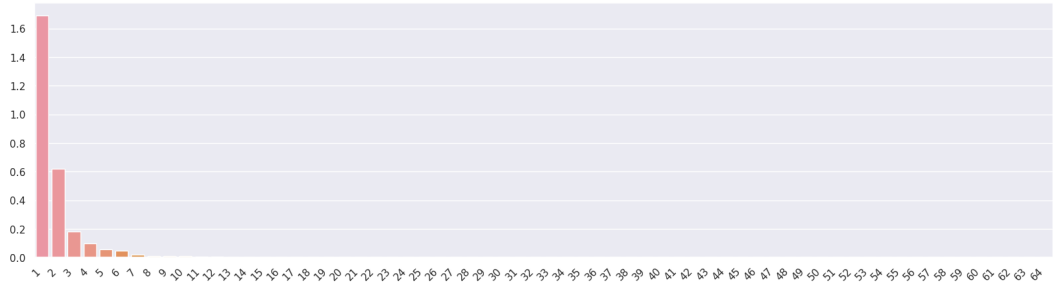
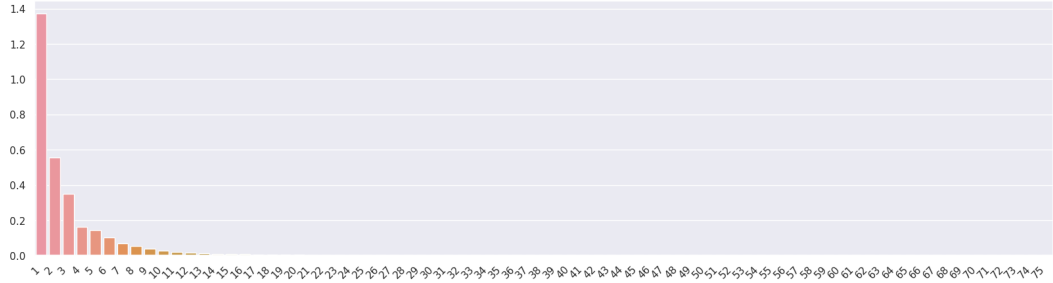


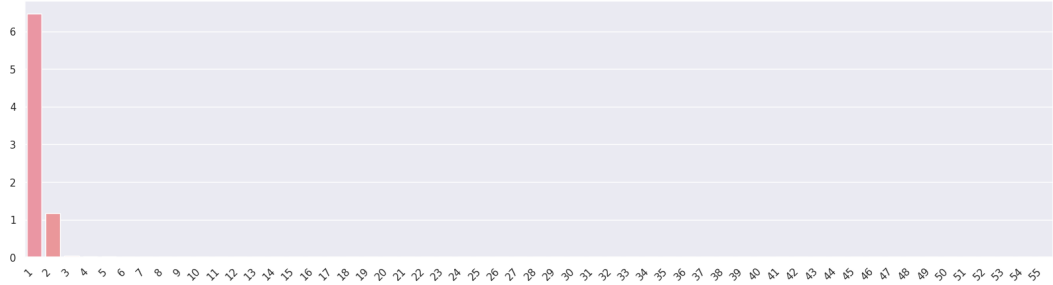
Figure 3: ■ : Training data ■ : Unseen data ■ : Inducing points
We query the above functions at $N = 1000$ equidistant points and add white noise with $\epsilon \sim \mathcal{N}(0, 0.5^2)$. We use $M = 30$ inducing points and train our method as described in Appendix A.7. The plot shows $m_Q(x) \pm 1.96\sqrt{\mathbb{V}[Y^*(x)|Y]}$ where $\mathbb{V}[Y^*(x)|Y]$ is the posterior predictive variance given as $r(x, x) + \sigma^2$. Here the fitted models from top to bottom are GWI-net, GWI-KRR, SVGP.



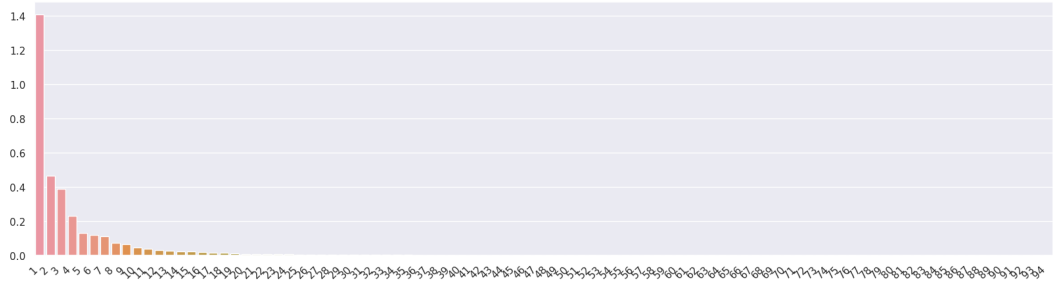
(a) BOSTON



(b) CONCRETE



(c) ENERGY



(d) WINE

(e) YACHT

Figure 4: The first 100 positive eigenvalues of $r(X_S, X)k(X, X_S)$ for datasets BOSTON, CONCRETE, ENERGY, WINE and YACHT.