Functional Stochastic Gradient MCMC for Bayesian Neural Networks

Mengjing Wu **Junyu Xuan** **Jie Lu**
Australian Artificial Intelligence Institute, University of Technology Sydney, Australia

Abstract

Classical parameter-space Bayesian inference for Bayesian neural networks (BNNs) suffers from several unresolved prior issues, such as knowledge encoding intractability and pathological behaviours in deep networks, which can lead to improper posterior inference. To address these issues, functional Bayesian inference has recently been proposed leveraging functional priors, such as the emerging functional variational inference. In addition to variational methods, stochastic gradient Markov Chain Monte Carlo (MCMC) is another scalable and effective inference method for BNNs to asymptotically generate samples from the true posterior by simulating continuous dynamics. However, existing MCMC methods perform solely in parameter space and inherit the unresolved prior issues, while extending these dynamics to function space is a non-trivial undertaking. In this paper, we introduce novel functional MCMC schemes, including stochastic gradient versions, based on newly designed diffusion dynamics that can incorporate more informative functional priors. Moreover, we prove that the stationary measure of these functional dynamics is the target posterior over functions. Our functional MCMC schemes demonstrate improved performance in both predictive accuracy and uncertainty quantification on several tasks compared to naive parameter-space MCMC and functional variational inference.

1 INTRODUCTION

Compared to traditional deep neural networks that rely on a single pattern of model parameters only, Bayesian neural networks (BNNs) demonstrate principled predictive uncertainty estimation and improved generalization by integrating models under the posterior distribution over network parameters (Neal, 1995; Gal et al., 2016; Wilson and Izmailov, 2020). As Bayesian inference methods continue to advance and gain success within the deep learning community, BNNs have been applied in a wide variety of domains and tasks, including dissolution prediction of planetary systems (Cranmer et al., 2021), medical diagnosis as diabetic detection (Filos et al., 2019; Band et al., 2021), classification of radio galaxies (Mohan and Scaife, 2024), and other safety-critical environments (Rudner et al., 2022).

Choosing an appropriate prior is critical for BNNs. The common practice is to use isotropy Gaussian priors for all network parameters, often referred to as parameter-space priors. However, these parameter-space priors have shown several problematic issues. For instance, as the depth of the network increases, the function samples of Gaussian prior over parameters tend to be horizontal (Duvenaud et al., 2014; Matthews et al., 2018). Additionally, with ReLU activations, the prior distribution of unit outputs becomes more heavy-tailed for deeper architectures.(Vladimirova et al., 2019; Tran et al., 2022). Furthermore, the effects of this parameter-space prior to posterior inference remain an open question due to the non-interpretability of network parameters and the complexity of network architectures Fortuin et al. (2022); Wild et al. (2022), which make it challenging to translate valid prior knowledge into a corresponding prior distribution over network parameters. Due to these unresolved issues with parameter-space priors, there has been growing interest in performing Bayesian inference directly in function space, where more informative stochastic processes can serve as priors. For example, Gaussian processes (GPs), which can easily encode prior knowledge about the potential functions via kernel functions, have been widely used as priors in functional variational inference for BNNs (Sun et al., 2019; Ma and Hernández-Lobato, 2021; Rudner et al., 2022; Pielok et al., 2022). These functional variational inference approaches have shown improved performance compared to parameter-space variational inference.

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

For posterior inference given a prior, Markov Chain Monte Carlo (MCMC) is another valid method for BNNs apart from variational inference. Unlike variational inference, which makes strong assumptions about the posterior distribution, non-parametric MCMC methods are asymptotically exact, making them the gold standard for Bayesian inference (Alexos et al., 2022). Even better, the computationally expensive MCMC methods became feasible for modern BNNs with large data following the introduction of their scalable stochastic gradient variants (SGMCMC) (Welling and Teh, 2011; Patterson and Teh, 2013; Chen et al., 2014; Zhang et al., 2020; Nemeth and Fearnhead, 2021). However, existing MCMC methods for BNNs typically perform in parameter space and are based on the diffusion dynamics designed in terms of the posterior over parameters, inheriting similar prior issues as parameter-space variational inference. Extending these methods to function space, however, is a non-trivial task. It requires the design of novel dynamics that not only sample effectively from the function space but also guarantee that the resulting stationary distribution corresponds to the target posterior over functions. This involves addressing the inherent complexities of functional spaces while ensuring the correctness and feasibility of the sampling process.

In this work, we propose novel functional MCMC schemes grounded in the fundamental Itô lemma for stochastic calculus (Itô, 1951; Brzeźniak et al., 2008; Øksendal, 2013). Specifically, the dynamics of BNNs in function space are designed by introducing new potential energy functionals that can incorporate functional priors, such as GPs, into the diffusion processes and simultaneously guarantee the target posterior over functions as the stationary measure. Our main contributions are as follows:

- We propose novel functional diffusion dynamics, along with their stochastic gradient versions for BNNs that effectively incorporate more informative functional priors into posterior inference. Additionally, we derive corresponding tractable MCMC samplers in parameter space, making them computationally feasible while maintaining the expressiveness of functional priors.
- We prove that the stationary measure would be the target posterior over functions under our designed functional Langevin dynamics and functional Hamiltonian dynamics.
- Our methods demonstrate improved predictive performance and uncertainty estimation in several benchmark tasks compared to parameter-space SGMCMC and parameter-/function-space variational inference methods.

2 PRELIMINARIES

2.1 Bayesian Neural Networks

Bayesian neural networks (BNNs) place probability distribution over network parameters rather than single fixed values used in standard neural networks. Given a dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^N = \{\mathbf{X}_\mathcal{D}, \mathbf{Y}_\mathcal{D}\}$, where $x_i \in \mathcal{X} \subseteq \mathbb{R}^p$ are the training inputs and $y_i \in \mathcal{Y} \subseteq \mathbb{R}^c$ are the corresponding targets, BNNs are stochastic neural networks characterized by random network parameters $\mathbf{w} \in \mathbb{R}^k$, treated as a k-dimensional multivariate random variable defined on a probability space $(\Omega, \mathcal{A}, P)$. The prior distribution over parameters is denoted by $p_0(\mathbf{w})$, and the likelihood evaluated on the training data is represented by the conditional distribution $p(\mathbf{Y}_\mathcal{D}|\mathbf{X}_\mathcal{D}; \mathbf{w})$. The posterior over parameters can then be inferred using Bayes' theorem, yielding the canonical form $p(\mathbf{w}|\mathcal{D}) \propto p(\mathbf{Y}_\mathcal{D}|\mathbf{X}_\mathcal{D}; \mathbf{w})p_0(\mathbf{w})$. Furthermore, the predictive distribution for test data $\{x^*, y^*\}$ is given by the expectation under the posterior as $p(y^*|x^*, \mathcal{D}) = \int p(y^*|x^*, \mathbf{w})p(\mathbf{w}|\mathcal{D})\mathrm{d}\mathbf{w}$. All notations used in this paper are listed in Table 5 in Appendix A.

2.2 Stochastic Gradient MCMC for BNNs

Dynamics-based sampling methods that leverage the gradients of the target log posterior can offer an efficient exploration of the parameter space (Ma et al., 2015; Izmailov et al., 2021; Li and Zhang, 2024). Furthermore, their stochastic gradient variants, such as the stochastic gradient Langevin dynamics (SGLD) (Welling and Teh, 2011) and the stochastic gradient Hamiltonian Monte Carlo (SGHMC) (Chen et al., 2014; Baker et al., 2019), are capable of scaling to large datasets for practical applications.

SGLD. The objective of MCMC for BNNs is to generate random samples from the posterior distribution over parameters (with the unnormalized form) as $p(\mathbf{w}|\mathcal{D}) \propto \exp(-U(\mathbf{w}))$, where $U(\mathbf{w}) = -\log p(\mathbf{Y}_\mathcal{D}|\mathbf{X}_\mathcal{D}; \mathbf{w}) - \log p_0(\mathbf{w})$ is the potential energy function. By combining Langevin dynamics (Neal et al., 2011) with the stochastic Robbins-Monro optimization algorithm (Robbins and Monro, 1951), Welling and Teh (2011) proposed SGLD as a scalable MCMC algorithm. The approximated discretization transition rule for posterior samples is

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \epsilon_t \nabla \tilde{U}(\mathbf{w}_t) + \sqrt{2\epsilon_t}\eta_t, \tag{1}$$

where ϵ_t denotes the step-size, and η_t is a standard Gaussian noise. The gradient $\nabla \tilde{U}(\mathbf{w})$ is computed using a mini-batch stochastic gradient for the likelihood term given by $\nabla \tilde{U}(\mathbf{w}) = -\frac{N}{n}\sum_{i=1}^n \nabla \log p(y_i|x_i; \mathbf{w}) - \nabla \log p(\mathbf{w})$ with $n \ll N$. As the step-size $\epsilon_t \to 0$, the

stochastic gradient noise vanishes more quickly than the injected Gaussian noise η_t, also ensuring that the discretization error is eliminated. Hence, the Metropolis-Hastings (MH) correction step can be ignored.

SGHMC. To enable more efficient exploration of the state space, Hamiltonian dynamics (Duane et al., 1987; Neal, 2012) introduces a momentum variable $\mathbf{z} \in \mathbb{R}^k$. The new target joint distribution is $p(\mathbf{w}|\mathcal{D})p(\mathbf{z}) \propto \exp(-H(\mathbf{w}, \mathbf{z}))$ with corresponding energy function $H(\mathbf{w}, \mathbf{z}) = -\log p(\mathbf{Y}_\mathcal{D}|\mathbf{X}_\mathcal{D}, \mathbf{w}) - \log p(\mathbf{w}) - \log p(\mathbf{z})$, where $p(\mathbf{z})$ is commonly assumed to be a zero-mean Gaussian $\mathcal{N}(\mathbf{z}|0, M)$. The sampling rule in the full-batch Hamiltonian Monte Carlo is defined as following with an MH step (Neal, 2012; Cobb and Jalaian, 2021):

$$\begin{aligned} \mathbf{w}_{t+1} &= \mathbf{w}_t + \epsilon_t M^{-1}\mathbf{z}_t, \\ \mathbf{z}_{t+1} &= \mathbf{z}_t - \epsilon_t \nabla U\left(\mathbf{w}_t\right). \end{aligned} \tag{2}$$

Like SGLD, SGHMC uses stochastic gradient $\nabla\tilde{U}(\mathbf{w})$ for the sampling of momentum $\mathbf{z}$, but with an additional friction term $\epsilon_t M^{-1}\mathbf{z}_t$ and a Gaussian injected noise to ensure that the (marginal) stationary distribution of the Hamiltonian dynamics remains the target posterior distribution (Chen et al., 2014). The specific update rule for the momentum $\mathbf{z}$ is given by $\mathbf{z}_{t+1} = \mathbf{z}_t - \epsilon_t\nabla\tilde{U}(\mathbf{w}_t) - \epsilon_t C M^{-1}\mathbf{z}_t + \sqrt{2C\epsilon_t}\eta_t$, where $C > 0$ is a scalar coefficient.

3 FUNCTIONAL DYNAMICS-BASED MCMC

In this section, we address the prior issues present in parameter-space MCMC by proposing two novel functional MCMC schemes based on newly designed diffusion dynamics, which allow for the incorporation of more informative functional priors. In Section 3.1, we introduce novel functional Langevin dynamics designed directly for the random function mapping defined by BNNs, along with a corresponding tractable stochastic gradient MCMC sampler for network parameters. We demonstrate that the stationary probability measure of this functional Langevin dynamics corresponds to the target posterior over functions. Additionally, in Section 3.2, we develop functional Hamiltonian dynamics also with a stochastic gradient version for BNNs. We prove that this scheme also guarantees the desired functional posterior as the stationary measure.

3.1 Functional Langevin Dynamics for BNNs

Dynamics-based MCMC techniques are grounded in the general framework of Itô diffusion (Øksendal, 2003). The main idea behind parameter-space MCMC methods for BNNs is to simulate a continuous diffusion process of network parameters $\mathbf{w}$ such that its stationary distribution is the target posterior $p(\mathbf{w}|\mathcal{D})$, using appropriately designed drift term and diffusion coefficient (see Appendix B). Naturally, to incorporate more informative functional prior into the posterior inference process, we now consider lifting the parameter-space Langevin dynamics onto function space.

Suppose $f(\cdot;\mathbf{w}) : \mathcal{X} \times \mathbb{R}^k \to \mathcal{Y}$ is the product-measurable random function mapping defined by a BNN. Assums $f(\cdot;\mathbf{w}) \in \mathbb{H}$, where $\mathbb{H}$ is an infinite-dimensional function space (separable Banach space) with Borel σ-algebra $\mathcal{B}(\mathbb{H})$. The likelihood is defined as p: $\mathcal{Y} \times \mathbb{H} \to [0, \infty)$, mapping $(\mathcal{Y}, f(\mathcal{X};\mathbf{w})) \mapsto p(\mathcal{Y}|f(\mathcal{X};\mathbf{w}))$, where $\mathcal{Y} \subseteq \mathbb{R}^c$ is Borel measurable and $p(\mathbf{Y}_\mathcal{D}|f(\mathbf{X}_\mathcal{D};\mathbf{w}))$ denotes the likelihood evaluated on the training data. Let $\mathcal{P}(\mathbb{H})$ represent the space of Borel probability measures on $\mathcal{B}(\mathbb{H})$, and assume the functional prior measure $P_0 \in \mathcal{P}(\mathbb{H})$ with topological support $\text{supp}(P_0)$, such as the classical GP prior denoted as $P_0 \sim \mathcal{GP}(\mathbf{m}, \mathbf{K})$. The posterior measure $P_{f|\mathcal{D}} \in \mathcal{P}(\mathbb{H})$, induced by the functional prior and likelihood, is defined by the Radon–Nikodym derivative as $P_{f|\mathcal{D}}(\mathrm{d}f) \propto \exp(-\Phi(f))P_0(\mathrm{d}f)$, where $\Phi(f)$ is the negative log-likelihood as $-\log p(\mathbf{Y}_\mathcal{D}|f(\mathbf{X}_\mathcal{D};\mathbf{w}))$ (Matthews et al., 2016; Lambley, 2023).

According to Itô Lemma (Itô, 1951), $f(\cdot;\mathbf{w})$ itself follows an Itô diffusion (Brzeźniak et al., 2008) (see Appendix B). Thus, we propose to directly simulate the diffusion process of the random function mapping $f(\cdot;\mathbf{w})$ of BNNs, where one can incorporate informative functional prior, P_0. Suppose $\varnothing \neq E \subseteq \text{supp}(P_0)$, and let $I_0 : E \to \mathbb{R}$ be an *Onsager–Machlup (OM) functional* for P_0, which can be heuristically interpreted as the negative logarithm of P_0 (Ayanbayev et al., 2021) (see Appendix B). Similar to parameter-space Langevin dynamics, the Langevin dynamics in function space can be generally represented by the stochastic differential equation (SDE): $\mathrm{d}f_t = -\nabla U(f_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$, where B is a standard Wiener process ((Brownian motion), and potential energy functional $U : E \to \mathbb{R}$ is given by $U(f) = \Phi(f) + I_0(f) = -\log p(\mathbf{Y}_\mathcal{D}|f(\mathbf{X}_\mathcal{D};\mathbf{w})) + I_0(f)$. This functional $U(f)$ is proven to be an OM functional for $P_{f|\mathcal{D}}$ (Lambley, 2023), ensuring that the target posterior measure $P_{f|\mathcal{D}}$ as the stationary measure $\pi(f)$. However, discretizing and sampling from this diffusion process in function space is intractable due to the infinite-dimensional nature of the problem.

Since $f(\cdot;\mathbf{w})$ is completely determined by the network parameters $\mathbf{w}$ (given a fixed network architecture), we address this challenge by designing a specific functional Langevin dynamics for $f(\cdot;\mathbf{w})$, which maintains the stationary measure as $P_{f|\mathcal{D}}$. We then transform this into the corresponding parameter-space dynamics for network parameters. We first design the following

functional Langevin dynamics for $f(\cdot;\mathbf{w})$ on $\mathbb{H}$:

$$\begin{aligned}&\mathrm{d}f_t(\cdot;\mathbf{w}) = \mu(f_t(\cdot;\mathbf{w}))\mathrm{d}t + \sigma(f_t(\cdot;\mathbf{w}))\mathrm{d}B_t\\&= \Big[-(\nabla_{\mathbf{w}} f_t)^T(\nabla_{\mathbf{w}} f_t)\nabla_f\big(-\log p(\mathbf{Y}_{\mathcal{D}}|f_t(\mathbf{X}_{\mathcal{D}};\mathbf{w})) + \\ &\quad I_0(f_t)\big) + H_{\mathbf{w}} f_t\Big]\mathrm{d}t + \sqrt{2}(\nabla_{\mathbf{w}} f_t)^T\mathrm{d}B_t,\end{aligned} \tag{3}$$

where $\mu(f(\cdot;\mathbf{w})) = -(\nabla_{\mathbf{w}} f)^T(\nabla_{\mathbf{w}} f)\nabla_f U(f) + H_{\mathbf{w}} f$ is the drift term, $\nabla_f U(f)$ denotes the Fréchet derivative of $U(f)$ at f, gradient $\nabla_{\mathbf{w}} f$ is the Fréchet derivative of f with respect to $\mathbf{w}$ (can be interpreted as a bounded operator from $\mathbb{R}^k$ to $\mathbb{H}$ for random functions), $H_{\mathbf{w}} f$ denotes the second-order Fréchet derivative of f. The diffusion coefficient $\sigma(f(\cdot;\mathbf{w}))$ is defined as $\sqrt{2}(\nabla_{\mathbf{w}} f)^T$, where B is a k-dimensional Brownian motion. The stationary probability measure of this dynamics is guaranteed to be $P_{f|\mathcal{D}}$ as stated in the following proposition.

Proposition 3.1. *The stationary probability measure of the functional Langevin dynamics defined in Equation* (3) *is the target posterior over functions* $P_{f|\mathcal{D}}$.

Proof. The proof is given in Appendix C. □

To establish a tractable functional MCMC scheme, we then transform the proposed functional Langevin dynamics for $f(\cdot;\mathbf{w})$ into the corresponding Langevin dynamics for the network parameters $\mathbf{w}$. Note that the differential of the function mapping $f(\cdot;\mathbf{w})$ is given by the Itô Lemma (Itô, 1951) as

$$\begin{aligned}&\mathrm{d}f_t(\cdot;\mathbf{w}) = (\nabla_{\mathbf{w}} f)^T\mathrm{d}\mathbf{w}_t + \frac{1}{2}(\mathrm{d}\mathbf{w}_t)^T(H_{\mathbf{w}} f)\mathrm{d}\mathbf{w}_t\\&= \left((\nabla_{\mathbf{w}} f)^T\mu(\mathbf{w}_t) + \frac{1}{2}Tr[\sigma(\mathbf{w}_t)^T(H_{\mathbf{w}} f)\sigma(\mathbf{w}_t)]\right)\mathrm{d}t\\&+ (\nabla_{\mathbf{w}} f)^T\sigma(\mathbf{w}_t)\mathrm{d}B_t,\end{aligned} \tag{4}$$

where $\mathrm{d}\mathbf{w}_t = \mu(\mathbf{w}_t)\mathrm{d}t + \sigma(\mathbf{w}_t)\mathrm{d}B_t$ is the general diffusion form for $\mathbf{w}$. By combining our functional Langevin dynamics defined in Equation (3) with this differential relationship between $f(\cdot;\mathbf{w})$ and $\mathbf{w}$, it is straightforward to reversely derive the corresponding diffusion process for $\mathbf{w}$ in parameter spaces as follows:

$$\begin{aligned}&\mathrm{d}\mathbf{w}_t = -\nabla_{\mathbf{w}} U(f_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t\\&= -\nabla_{\mathbf{w}} f_t\big[\nabla_f\big(-\log p(\mathbf{Y}_{\mathcal{D}}|f_t(\mathbf{X}_{\mathcal{D}};\mathbf{w})) + I_0(f_t)\big)\big]\mathrm{d}t\\&+ \sqrt{2}\mathrm{d}B_t,\end{aligned} \tag{5}$$

where the drift term $\mu(\mathbf{w}) = -\nabla_{\mathbf{w}} U(f)$, and the diffusion coefficient $\sigma(\mathbf{w}) = \sqrt{2}\cdot\mathbf{I}_k$. Then, by discretizing and sampling from this corresponding diffusion process for the network parameters, we can obtain function samples from the target posterior measure $P_{f|\mathcal{D}}$ in a tractable manner.

In practice, the stochastic gradient variant for the drift term of SDE in Equation (5) is denoted as $-\nabla_{\mathbf{w}}\tilde{U}(f)$, which utilizes the minibatch computation for the likelihood term. To estimate the gradient of the prior measure, we assume an approximate GP prior for BNNs. This is motivated by the fact that, in the limit of infinite width, the prior measure of BNNs converges to a GP with the Neural Network Gaussian Process (NNGP) kernel (Neal, 1995; Lee et al., 2018; Matthews et al., 2018; Lee et al., 2019). Due to the infinite-dimensional nature of the functional distribution, we solve the gradient of functional prior on finite measurement points $\mathbf{X}_{\mathcal{M}} \overset{\text{def}}{=} [\mathbf{x}_1,\ldots,\mathbf{x}_M]^{\mathrm{T}}$ as $\nabla_{f^{\mathbf{X}_{\mathcal{M}}}}\log P_0(f^{\mathbf{X}_{\mathcal{M}}})$, where $f^{\mathbf{X}_{\mathcal{M}}}$ are corresponding function values evaluated at $\mathbf{X}_{\mathcal{M}}$. The prior $P_0(f^{\mathbf{X}_{\mathcal{M}}})$ reduces to a multivariate Gaussian, allowing for an analytical solution for the gradient. The specific discretization update rule for network parameters samples under functional stochastic gradient Langevin dynamics (fSGLD) is given by

$$\begin{aligned}\mathbf{w}_{t+1} =& \mathbf{w}_t - \epsilon_t\nabla_{\mathbf{w}} f_t\Big[-\frac{N}{n}\sum_{i=1}^{n}\nabla_f\log p(y_i|f_t(x_i;\mathbf{w}_t)) - \\ &\nabla_{f^{\mathbf{X}_{\mathcal{M}}}}\log P_0(f_t^{\mathbf{X}_{\mathcal{M}}})\Big] + \sqrt{2\epsilon_t}\eta_t.\end{aligned} \tag{6}$$

Like the standard SGLD, with the discretization stepsize ϵ_t decays to zero, the stationary measure of fSGLD continues to be the target posterior $P_{f|\mathcal{D}}$ over functions.

Note that in contrast to the naive parameter-space SGLD in Equation (1), which is driven solely by limited parameter information, the discretization update rule for samples of network parameters in Equation (6) under our fSGLD scheme incorporates information about the function mapping. This additional functional information plays a crucial role in achieving more accurate posterior inference. The pseudocode for fSGLD is shown in Algorithm 1 in Appendix D.

3.2 Functional Hamiltonian Dynamics for BNNs

For the naive Hamiltonian dynamics, $X = (\mathbf{w},\mathbf{z})$ represents the augmentation of model parameters. We now let $X = (f,g)$ and define the Hamiltonian dynamics in function space as $H(f,g) = U(f) - \log p(g)$, where $p(g)$ is the functional auxiliary probability measure induced by the distribution of auxiliary variable $\mathbf{z}$. We then design the functional Hamiltonian dynamics for $f(\cdot;\mathbf{w})$ and $g(\cdot;\mathbf{z})$ as

$$\begin{aligned}&\mathrm{d}f_t(\cdot;\mathbf{w}) = -(\nabla_{\mathbf{w}} f_t)^T\nabla_{\mathbf{z}} g_t\nabla_g\log p(g_t)\mathrm{d}t,\\&\mathrm{d}g_t(\cdot;\mathbf{z}) = -(\nabla_{\mathbf{z}} g_t)^T\nabla_{\mathbf{w}} f_t\nabla_f U(f_t)\mathrm{d}t.\end{aligned} \tag{7}$$

Proposition 3.2. *The (marginal) stationary probability measure of the functional Hamiltonian dynamics*

Table 1: Drift term and diffusion coefficient for naive SGMCMC and functional SGMCMC algorithms.

DYNAMICS	$\mu(\cdot)$	$\sigma(\cdot)$
SGLD	$-\nabla\tilde{U}(\mathbf{w})$	$\sqrt{2}\cdot\mathbf{I}_k$
FSGLD	$-(\nabla_{\mathbf{w}}f)^T(\nabla_{\mathbf{w}}f)\nabla_f\tilde{U}(f)+H_{\mathbf{w}}f$	$\sqrt{2}(\nabla_{\mathbf{w}}f)^T$
SGHMC	$-\begin{pmatrix}\mathbf{0} & -\mathbf{I}_k\\ \mathbf{I}_k & C\end{pmatrix}\begin{pmatrix}\nabla\tilde{U}(\mathbf{w})\\ M^{-1}\mathbf{z}\end{pmatrix}$	$\begin{pmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \sqrt{2C}\end{pmatrix}$
FSGHMC	$-\begin{pmatrix}\mathbf{0} & -(\nabla_{\mathbf{w}}f)^T\nabla_{\mathbf{z}}g\\ (\nabla_{\mathbf{z}}g)^T\nabla_{\mathbf{w}}f & C(\nabla_{\mathbf{z}}g)^T\nabla_{\mathbf{z}}g\end{pmatrix}\begin{pmatrix}\nabla_f\tilde{U}(f)\\ -\nabla_g\log p(g)\end{pmatrix}$	$\begin{pmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \sqrt{2C}(\nabla_{\mathbf{z}}g)^T\end{pmatrix}$

defined in Equation (7) *is the target posterior over functions* $P_{f|\mathcal{D}}$.

Proof. The proof is given in Appendix C. □

Similar to the fSGLD, we also transform the above function-space Hamiltonian dynamics to its corresponding parameter-space dynamics of $\mathbf{w}$ and $\mathbf{z}$ as follows:

$$\begin{aligned}\mathrm{d}\mathbf{w}_t &= \frac{\partial H}{\partial \mathbf{z}} = -\nabla_{\mathbf{z}}g_t\cdot\nabla_g\log p(g_t)\mathrm{d}t,\\ \mathrm{d}\mathbf{z}_t &= -\frac{\partial H}{\partial \mathbf{w}} = -\nabla_{\mathbf{w}}f_t\cdot\nabla_f U(f_t)\mathrm{d}t,\end{aligned} \tag{8}$$

where the random diffusion term is 0 for the full-batch version.

For the stochastic gradient version, the discretization update rule for samples of $\mathbf{w}$ and $\mathbf{z}$ under the functional stochastic gradient Hamiltonian dynamics (fSGHMC) is as follows:

$$\begin{aligned}\mathbf{w}_{t+1} &= \mathbf{w}_t - \epsilon_t\nabla_{\mathbf{z}}g_t\cdot\nabla_g\log p(g_t)\\ \mathbf{z}_{t+1} &= \mathbf{z}_t - \epsilon_t\nabla_{\mathbf{w}}f_t\cdot\nabla_f\tilde{U}(f_t)\\ &\quad + \epsilon_t C\nabla_{\mathbf{z}}g_t\cdot\nabla_g\log p(g_t) + \sqrt{2C\epsilon_t}\eta_t.\end{aligned} \tag{9}$$

The stationary probability measure of such functional stochastic gradient Hamiltonian can still be guaranteed as the target functional posterior $P_{f|\mathcal{D}}$ (see Appendix C). Like the discussion in the last section, the diffusion process for $\mathbf{w}$ and $\mathbf{z}$ under fSGHMC scheme in Equation (9) incorporates abundant functional information is obviously different from the naive parameter-space SGHMC in Equation (2). The pseudocode for fSGHMC is shown in Algorithm 2 in Appendix D.

For our functional MCMC schemes, the predictive distribution is obtained from the following integration process:

$$\begin{aligned}p(y^*|x^*,\mathcal{D}) &= \int p(y^*|f(x^*))P_{f|\mathcal{D}}\mathrm{d}f\\ &\approx \frac{1}{S}\sum_{j=1}^{S}p(y^*|f(x^*;\mathbf{w}^{(j)})),\end{aligned} \tag{10}$$

where function draws of posterior over functions are generated from corresponding samples of network parameters $\mathbf{w}^{(j)}(j=1,2,...,S)$ using fSGLD or fSGHMC.

The comparisons of the definitions for the drift term and the diffusion coefficient of popular exiting SGMCMC algorithms and our functional fSGLD and fSGHMC are summarized in Table 1.

4 RELATED WORK

Functional Variational Inference for BNNs. To specify meaningful functional priors like Gaussian processes, Sun et al. (2019) proposed a functional evidence lower bound (ELBO) to optimize the Kullback–Leibler (KL) divergence between variational posterior and the true posterior in function space explicitly. They solved the KL divergence between infinite-dimensional stochastic processes on finite marginal measurement sets. Due to the analytical intractability of the supremum over marginal KL divergences in functional ELBO, Rudner et al. (2022) proposed to approximate the posterior and prior over functions of BNNs as Gaussian distributions through local linearization and derived a more tractable functional variational objective for BNNs. Considering the potential weaknesses of KL divergence for stochastic processes (Burt et al., 2020), Tran et al. (2022) proposed to match a BNN prior to a GP prior by minimizing the 1-Wasserstein distance (Kantorovich, 1960) to obtain a more interpretable functional prior.

Markov Chain Monte Carlo for BNNs. Papamarkou et al. (2022) summarized the challenges in applying MCMC approaches to posterior inference for BNNs in four aspects: computational cost, model structure, weight symmetries, and prior specification. MCMC methods had limited success in being broadly adopted for posterior inference in large modern BNNs until the introduction of scalable Monte Carlo algorithms, such as the stochastic gradient MCMC (Welling and Teh, 2011; Chen et al., 2014; Ahn et al., 2014; Garriga-Alonso and Fortuin, 2020). Ma et al. (2015) designed a unifying framework that casts all existing dynamics-based samplers and their minibatch variants into it. Regarding the multimodal properties of poste-

rior distributions, Zhang et al. (2020) proposed a cyclical step-size schedule in stochastic gradient MCMC for BNNs to improve the sampling efficiency. On the other hand, Cobb and Jalaian (2021) emphasized the limitations of stochastic gradients in HMC and proposed a scaling full-batch HMC for BNNs using a symmetric splitting integration scheme. However, all these MCMC methods are performed in parameter space.

5 EXPERIMENTS

We evaluate the predictive performance and uncertainty quantification of our functional stochastic gradient MCMC schemes on several benchmark tasks, including a synthetic extrapolation example, multivariate regressions on UCI datasets, image classification tasks, and contextual bandits. We compare fSGLD and fSGHMC with naive SGLD, SGHMC, and competing functional variational inference methods to illustrate the sampling efficiency of our functional stochastic gradient MCMC schemes [1].

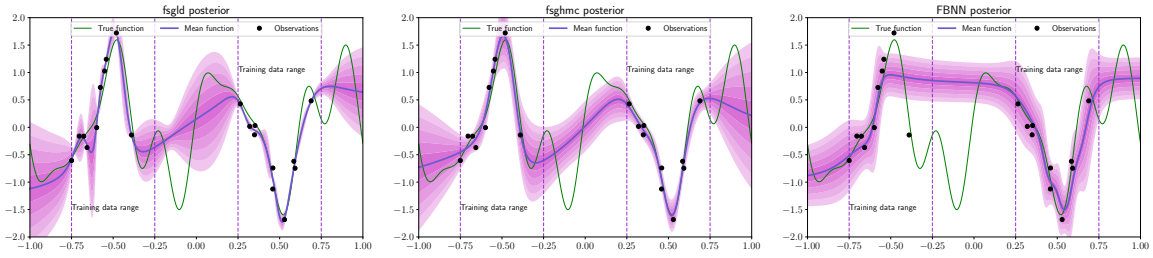
5.1 Extrapolation on Synthetic Data

To evaluate the fitting ability and uncertainty estimation, we first consider a 1-D oscillation curve extrapolation on a synthetic dataset. For input, we randomly sampled half of 20 observation points from Uniform$(-0.75, -0.25)$, and the other half are from Uniform$(0.25, 0.75)$. The output is modeled through the polynomial function: $y = \sin(3\pi x) + 0.3\cos(9\pi x) + 0.5\sin(7\pi x) + \epsilon$ with noise $\epsilon \sim \mathcal{N}(0, 0.5^2)$. We compared our fSGLD and fSGHMC with naive SGLD, SGHMC, and two representative functional variational methods: FBNN (Sun et al., 2019) and IFBNN (Wu et al., 2023). The model is a fully-connected neural network with two hidden layers. For our functional fSGLD, fSGHMC, and the two functional variational inference methods, we use the same functional GP prior with the RBF kernel pre-trained on the input dataset. There are 40 measurement points for the approximate estimation of the gradient of log functional prior in fSGLD and fSGHMC, which are randomly sampled from training data and an additional 40 inducing points from Uniform $(-1, 1)$. For all sampling methods, we use 2000 burn-in iterations and 8000 iterations for 80 samples (to reduce correlations between samples, we draw separated samples every 100 epochs). Functional variational inference methods are trained for 10000 epochs for fair comparison. Results are shown in Figure 1. Figure 1(a) and Figure 1(b) show that fSGLD and fSGHMC can recover the key characteristic of the curve in the observation range and provide reasonable uncertainty estimations in the unseen region. fSGHMC performs slightly better than fSGLD. In contrast, for naive SGMCMC methods, Figure 1(d) illustrates that SGLD underestimates the uncertainty in non-observation areas, especially evident in the leftmost non-observed $[-1, -0.75]$ interval. Meanwhile, Figure 1(e) shows that SGHMC suffers from some unreasonable manifestations of the uncertainty estimation, such as the undeserved uncertainty expansion in the interval $[-0.4, -0.25]$ of the input data range. The improved performances of our fSGLD and fSGHMC can be attributed to two key factors: (i) Incorporation of more meaningful prior information. The functional GP prior with an RBF kernel is well-suited for modelling polynomial functions, effectively incorporating rich prior knowledge about the underlying functions. In contrast, SGLD and SGHMC rely on an isotropic Gaussian prior over network parameters, which makes it difficult to encode polynomial structures. Moreover, such parameter-space prior, often deemed uninformative, can introduce incorrect assumptions about the posterior (Cinquin et al., 2024), such as uni-modality and parameter independence, leading to suboptimal posterior inference; (ii) Exploration in function space. Our functional dynamics are explored directly in function space. SGLD and SGHMC sampling in the highly multi-modal parameter space, where the target posterior over parameters often contains numerous distant local modes that correspond to the same function, making exploration inefficient and complex. By contrast, our functional methods facilitate a more efficient exploration of the target posterior over functions, yielding more continuous exploration paths across the energy landscape in function space. On the other hand, the rightmost column shows results from two functional variational methods. We found that FBNN in Figure 1(c) is unable to fit the target curve in the training data range $[-0.75, 0.25]$ and performs poorly for uncertainty quantification. Figure 1(f) shows that IFBNN also severely underestimates the predictive uncertainty, which is consistent with the findings of Ormerod and Wand (2010) and Zhang et al. (2018) that variational approximation typically underestimates the posterior variance. More analysis of mixing rate and computational complexity are shown in Appendix F and Appendix G, respectively. Appendix H presents a wide ablation study about the sample size, measurement points and functional prior.

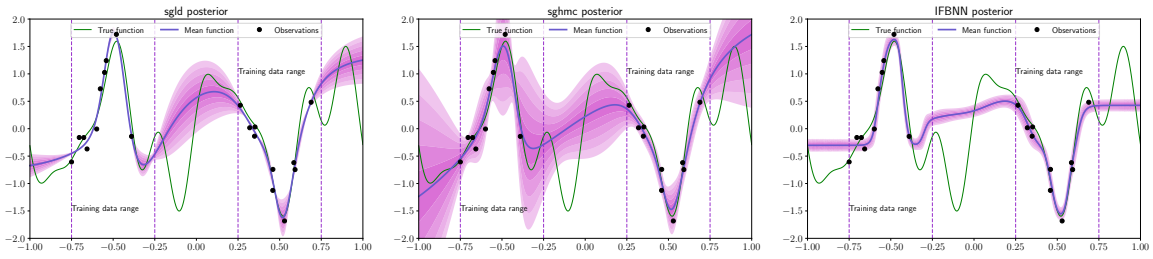
5.2 UCI Regression

In this experiment, we evaluate our methods on multivariate regression tasks across seven benchmark UCI datasets: *Yacht, Boston, Concrete, Energy, Wine, Kin8nm and Protein*. We use a two-hidden-layer fully connected neural network. For all sampling methods, we run 500 iterations for the burn-in stage and collect 15 samples. Performance is measured via average root

[1] Codebase: https://github.com/Mengjinguts/fsgmcmc



(a) fSGLD (b) fSGHMC (c) FBNN



(d) SGLD (e) SGHMC (f) IFBNN

Figure 1: 1-D extrapolation example. The green line is the ground true function, and the blue lines correspond to the mean of samples from posterior predictions. Black dots denote 20 training points; shadow areas represent the predictive standard deviations. For more details, see Appendix E.

mean square error (RMSE) and the test negative log-likelihood (NLL). For each dataset, we construct 10 random 90-to-10 train-test splits and report the mean and standard deviation of RMSE and NLL over these splits in Table 2 and Table 3, respectively. fSGHMC outperforms all baselines for RMSE on all datasets, and fSGLD also achieves better accuracy values than naive SGMCMC methods and functional variational inference, demonstrating superior performance of our functional SGMCMC in predictive accuracy. Moreover, fSGHMC and fSGLD obtain better NLL results on most of the datasets (5/7), showing that our methods are also very competitive in uncertainty quantification.

5.3 Imagine Classification and OOD Detection

In this section, we investigate the scalability of our methods on image classification tasks with high-dimensional inputs. We evaluate the in-distribution predictive performance and out-of-distribution (OOD) detection ability on MNIST (LeCun et al., 2010), Fashion-MNIST(Xiao et al., 2017) and CIFAR-10(Krizhevsky et al., 2009). We use ResNet-18 architecture (He et al., 2016) for all methods on these three datasets, and the batch size is 125. For all sampling methods, we run for 60 burn-in iterations and collect 10 samples on the MNIST dataset, 100 burn-in iterations and collect 10 samples on FashionMNIST, 80 burn-in iterations and draw 10 samples on CIFAR10. We randomly sample 40 points from training data in every iteration as the measurement points for the gradient estimation of log functional prior distribution in fSGLD and fSGHMC. We report the test classification errors for predictive performance and the area under the curve (AUC) of OOD detection pairs FashionMNIST/MNIST, MNIST/FashionMNIST and CIFAR10/SVNH based on predictive entropies in Table 4. Our fSGLD and fSGHMC outperform all other methods for both predictive accuracy and OOD detection.

5.4 Contextual Bandits

In many downstream tasks, such as online sequential decision-making scenarios in contextual bandits problems, reliable uncertainty estimation is important to guide the exploration-exploitation process. In such problems, an agent interacts with an unknown environment repeatedly and chooses to take an optimal action in each round of interaction. Thompson sampling (Thompson, 1933; Russo and Van Roy, 2016) is a widely used algorithm for exploration strategies in contextual bandits, it (i) first samples a current posterior to obtain a model configuration; (ii) then adaptively chooses an optimal action based on the current context under the sampled model configuration and observes the corresponding reward; (iii) updates the posterior based on the context, action and reward tuples in this

Table 2: The table shows the results of average RMSE for multivariate regression on UCI datasets. We split each dataset randomly into 90% training data and 10% test data, and this process is repeated 10 times to ensure validity. Bold indicates statistically significant best results ($p < 0.01$ with t-test).

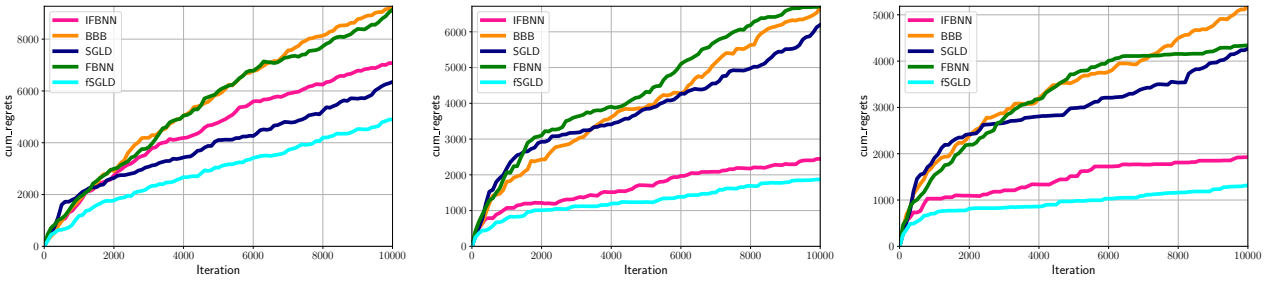
	RMSE					
	SGLD	fSGLD	SGHMC	fSGHMC	FBNN	IFBNN
Yacht	1.09 ± 0.10	0.41 ± 0.10	1.19 ± 0.10	**0.25 ± 0.13**	1.52 ± 0.08	1.24 ± 0.10
Boston	1.23 ± 0.06	0.36 ± 0.09	1.31 ± 0.06	**0.24 ± 0.10**	1.68 ± 0.12	1.44 ± 0.09
Concrete	1.10 ± 0.07	0.45 ± 0.04	1.15 ± 0.06	**0.28 ± 0.05**	1.27 ± 0.05	1.07 ± 0.07
Energy	1.04 ± 0.06	0.24 ± 0.03	1.07 ± 0.08	**0.18 ± 0.03**	1.35 ± 0.06	1.19 ± 0.05
Wine	1.08 ± 0.09	0.71 ± 0.05	1.14 ± 0.09	**0.62 ± 0.04**	1.53 ± 0.05	1.21 ± 0.05
Kin8nm	1.20 ± 0.02	0.74 ± 0.02	1.26 ± 0.03	**0.37 ± 0.01**	1.45 ± 0.07	1.12 ± 0.02
Protein	1.12 ± 0.01	0.84 ± 0.01	1.21 ± 0.01	**0.83 ± 0.01**	1.50 ± 0.03	1.16 ± 0.01

Table 3: The table shows the results of average NLL for multivariate regression on UCI datasets. We split each dataset randomly into 90% training data and 10% test data, and this process is repeated 10 times to ensure validity. Bold indicates statistically significant best results ($p < 0.01$ with t-test).

	NLL					
	SGLD	fSGLD	SGHMC	fSGHMC	FBNN	IFBNN
Yacht	-0.37 ± 0.08	-2.46 ± 0.28	-2.06 ± 0.10	**-3.37 ± 0.69**	-0.77 ± 0.86	-1.25 ± 1.21
Boston	-0.60 ± 0.11	-2.20 ± 0.20	-2.25 ± 0.08	**-3.26 ±0.33**	-1.19 ± 0.76	0.32 ± 0.30
Concrete	-0.54 ± 0.05	-1.63 ± 0.14	-2.51 ± 0.08	**-2.74 ±0.18**	-1.00 ± 0.52	-0.39 ± 0.33
Energy	-0.58 ± 0.09	-2.25 ± 0.25	-2.39 ± 0.09	**-4.29 ±0.23**	-2.14 ± 0.47	-1.78 ± 0.36
Wine	-0.90 ± 0.12	**-2.53 ± 0.15**	-2.50 ± 0.10	-2.34 ± 0.25	0.52 ± 0.14	0.26 ± 0.15
Kin8nm	-0.60 ± 0.04	-2.24 ± 0.09	-2.29 ± 0.04	-2.00 ±0.12	**-2.45 ± 0.62**	-1.01 ± 0.14
Protein	-0.60 ± 0.01	-2.10 ± 0.08	**-2.20 ± 0.01**	-2.02 ±0.06	-1.49 ± 0.24	-2.13 ± 0.34

Table 4: Image classification and OOD detection performance.

	MNIST		FMNIST		CIFAR10	
Model	Test Error	AUC	Test Error	AUC	Test Error	AUC
SGLD	2.73 ± 0.00	0.859 ± 0.06	13.92 ± 0.01	0.633 ± 0.13	49.56 ± 0.03	0.669 ± 0.04
fSGLD	1.89 ± 0.00	**0.960 ± 0.02**	**11.66 ± 0.01**	0.849 ± 0.03	33.57 ± 0.02	0.672 ± 0.03
SGHMC	2.97 ± 0.00	0.853 ± 0.05	13.72 ± 0.01	0.589 ± 0.07	48.13 ± 0.04	0.639 ± 0.06
fSGHMC	**1.64 ± 0.00**	0.931 ± 0.02	11.91 ± 0.00	**0.880 ± 0.03**	**31.99 ± 0.01**	**0.717 ± 0.03**
FBNN	3.99 ± 0.00	0.801 ± 0.03	14.36 ± 0.00	0.814 ± 0.02	53.71 ± 0.01	0.612 ± 0.03
IFBNN	3.64 ± 0.00	0.949 ± 0.03	14.15 ± 0.00	0.838 ± 0.01	53.38 ± 0.01	0.616 ± 0.03



(a) p = 0.4 (b) p = 0.5 (c) p = 0.6

Figure 2: Comparisons of cumulative regrets of fSGLD, SGLD, FBNN, IFBNN, BBB for contextual bandit task on the Mushroom dataset. Lower represents better performance.

interaction round.

This section evaluates the ability to guide exploration on the UCI Mushroom dataset with 8124 instances. In each instance, the mushroom has 22 features and is labeled as edible or poisonous. An agent can observe mushroom features as the context in each interaction and choose to eat or reject a mushroom. We compare our fSGLD with several baselines, including naive SGLD, functional variational FBNN and IFBNN, parametric-space variational method Bayes By Backprop (BBB) (Blundell et al., 2015). We follow the basic

settings by Blundell et al. (2015) and consider three different reward patterns: if the agent eats an edible mushroom, it receives a reward of 5 and a reward of 0 if the agent chooses to reject. On the other hand, if the agent eats a poisonous mushroom, it receives a reward of -35 with probabilities of 0.4, 0.5, and 0.6, respectively, for three different patterns. If the agent chooses to reject a poisonous mushroom, it receives a reward of 0. Suppose an oracle always chooses to eat an edible mushroom and reject the poisonous mushroom. The cumulative regrets with respect to the reward achieved by the oracle measure the exploration-exploitation ability of an agent. We run 10000 iterations for all methods and the minibatch size is 64. Figure 2 shows the cumulative regrets for all methods in three reward patterns. Our fSGLD consistently outperforms other inference methods, indicating that our functional SGMCMC can provide reliable uncertainty estimation in decision-making tasks.

6 DISCUSSION AND CONCLUSION

In this paper, we lift the Itô diffusion of network parameters in BNNs onto function space and propose novel functional MCMC schemes for BNNs. Specifically, we design functional Langevin dynamics and functional Hamiltonian dynamics for the function mapping defined by BNNs and derive corresponding tractable sampling methods in parameter space, which can incorporate informative functional prior into posterior inference. Moreover, we proved that the stationary measure of our functional dynamics is the posterior over functions. Empirically, we demonstrate that our functional MCMC schemes effectively leverage functional priors to yield superior predictive performance and principled uncertainty quantification on several benchmark tasks.

Acknowledgements

This work is supported by the Australian Research Council under the Discovery Early Career Researcher Award DE200100245.

References

Ahn, S., Shahbaba, B., and Welling, M. (2014). Distributed stochastic gradient mcmc. In *International conference on machine learning*, pages 1044–1052. PMLR.

Alexos, A., Boyd, A. J., and Mandt, S. (2022). Structured stochastic gradient mcmc. In *International Conference on Machine Learning*, pages 414–434. PMLR.

Ayanbayev, B., Klebanov, I., Lie, H. C., and Sullivan, T. J. (2021). Γ-convergence of onsager–machlup functionals: II. infinite product measures on banach spaces. *Inverse Problems*, 38(2):025006.

Baker, J., Fearnhead, P., Fox, E. B., and Nemeth, C. (2019). Control variates for stochastic gradient mcmc. *Statistics and Computing*, 29:599–615.

Band, N., Rudner, T. G., Feng, Q., Filos, A., Nado, Z., Dusenberry, M. W., Jerfel, G., Tran, D., and Gal, Y. (2021). Benchmarking bayesian deep learning on diabetic retinopathy detection tasks. In *35th Conference on Neural Information Processing Systems*.

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

Brzeźniak, Z., van Neerven, J. M., Veraar, M. C., and Weis, L. (2008). Itô's formula in umd banach spaces and regularity of solutions of the zakai equation. *Journal of Differential Equations*, 245(1):30–58.

Burt, D. R., Ober, S. W., Garriga-Alonso, A., and van der Wilk, M. (2020). Understanding variational inference in function-space. In *Third Symposium on Advances in Approximate Bayesian Inference*, pages 1–17.

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

Cinquin, T., Pförtner, M., Fortuin, V., Hennig, P., and Bamler, R. (2024). FSP-laplace: Function-space priors for the laplace approximation in bayesian deep learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Cobb, A. D. and Jalaian, B. (2021). Scaling hamiltonian monte carlo inference for bayesian neural networks with symmetric splitting. In *Uncertainty in Artificial Intelligence*, pages 675–685. PMLR.

Cranmer, M., Tamayo, D., Rein, H., Battaglia, P., Hadden, S., Armitage, P. J., Ho, S., and Spergel, D. N. (2021). A bayesian neural network predicts the dissolution of compact planetary systems. *Proceedings of the National Academy of Sciences*, 118(40):e2026053118.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid monte carlo. *Physics letters B*, 195(2):216–222.

Duvenaud, D., Rippel, O., Adams, R., and Ghahramani, Z. (2014). Avoiding pathologies in very deep networks. In *Artificial Intelligence and Statistics*, pages 202–210. PMLR.

Filos, A., Farquhar, S., Gomez, A. N., Rudner, T. G., Kenton, Z., Smith, L., Alizadeh, M., De Kroon,

A., and Gal, Y. (2019). A systematic comparison of bayesian deep learning robustness in diabetic retinopathy tasks. In *4th workshop on Bayesian Deep Learning (NeurIPS 2019)*.

Fortuin, V., Garriga-Alonso, A., Ober, S. W., Wenzel, F., Ratsch, G., Turner, R. E., van der Wilk, M., and Aitchison, L. (2022). Bayesian neural network priors revisited. In *International Conference on Learning Representations*.

Gal, Y. et al. (2016). Uncertainty in deep learning.

Garriga-Alonso, A. and Fortuin, V. (2020). Exact langevin dynamics with stochastic gradients. In *3rd Symposium on Advances in Approximate Bayesian Inference*, pages 1–10.

Ghosh, A. P. (2010). Backward and forward equations for diffusion processes. *Wiley Encyclopedia of Operations Research and Management Science*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Itô, K. (1951). On a formula concerning stochastic differentials. *Nagoya Mathematical Journal*, 3:55–65.

Izmailov, P., Vikram, S., Hoffman, M. D., and Wilson, A. G. G. (2021). What are bayesian neural network posteriors really like? In *International conference on machine learning*, pages 4629–4640. PMLR.

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

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

Lambley, H. (2023). Strong maximum a posteriori estimation in banach spaces with gaussian priors. *Inverse Problems*, 39(12):125010.

LeCun, Y., Cortes, C., Burges, C., et al. (2010). Mnist handwritten digit database.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. (2018). Deep neural networks as gaussian processes. In *International Conference on Learning Representations*.

Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. (2019). Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32.

Li, B. and Zhang, R. (2024). Entropy-mcmc: Sampling from flat basins with ease. In *International Conference on Learning Representations*.

Ma, C. and Hernández-Lobato, J. M. (2021). Functional variational inference based on stochastic process generators. *Advances in Neural Information Processing Systems*, 34:21795–21807.

Ma, Y.-A., Chen, T., and Fox, E. (2015). A complete recipe for stochastic gradient mcmc. *Advances in neural information processing systems*, 28.

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

Matthews, A. G. d. G., Hron, J., Rowland, M., Turner, R. E., and Ghahramani, Z. (2018). Gaussian process behaviour in wide deep neural networks. In *International Conference on Learning Representations*.

Mohan, D. and Scaife, A. M. M. (2024). Evaluating bayesian deep learning for radio galaxy classification. In *The 40th Conference on Uncertainty in Artificial Intelligence*.

Neal, R. M. (1995). *BAYESIAN LEARNING FOR NEURAL NETWORKS*. PhD thesis, University of Toronto.

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

Neal, R. M. et al. (2011). Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2.

Nemeth, C. and Fearnhead, P. (2021). Stochastic gradient markov chain monte carlo. *Journal of the American Statistical Association*, 116(533):433–450.

Øksendal, B. (2003). *Stochastic differential equations*. Springer.

Øksendal, B. (2013). *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media.

Ormerod, J. T. and Wand, M. P. (2010). Explaining variational approximations. *The American Statistician*, 64(2):140–153.

Papamarkou, T., Hinkle, J., Young, M. T., and Womble, D. (2022). Challenges in markov chain monte carlo for bayesian neural networks. *Statistical Science*, 37(3):425–442.

Patterson, S. and Teh, Y. W. (2013). Stochastic gradient riemannian langevin dynamics on the probability simplex. *Advances in neural information processing systems*, 26.

Pielok, T., Bischl, B., and Rügamer, D. (2022). Approximate bayesian inference with stein functional variational gradient descent. In *The Eleventh International Conference on Learning Representations*.

Risken, H. (1996). *Fokker-planck equation*. Springer.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Rudner, T. G., Chen, Z., Teh, Y. W., and Gal, Y. (2022). Tractable function-space variational inference in bayesian neural networks. *Advances in Neural Information Processing Systems*, 35:22686–22698.

Russo, D. and Van Roy, B. (2016). An information-theoretic analysis of thompson sampling. *The Journal of Machine Learning Research*, 17(1):2442–2471.

Shi, J., Chen, T., Yuan, R., Yuan, B., and Ao, P. (2012). Relation of a new interpretation of stochastic differential equations to ito process. *Journal of Statistical physics*, 148:579–590.

Sun, S., Zhang, G., Shi, J., and Grosse, R. (2019). Functional variational bayesian neural networks. In *International Conference on Learning Representations*.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294.

Tran, B.-H., Rossi, S., Milios, D., and Filippone, M. (2022). All you need is a good functional prior for bayesian deep learning. *The Journal of Machine Learning Research*, 23(1):3210–3265.

Vladimirova, M., Verbeek, J., Mesejo, P., and Arbel, J. (2019). Understanding priors in bayesian neural networks at the unit level. In *International Conference on Machine Learning*, pages 6458–6467. PMLR.

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

Wild, V. D., Hu, R., and Sejdinovic, D. (2022). Generalized variational inference in function spaces: Gaussian measures meet bayesian deep learning. *Advances in Neural Information Processing Systems*, 35:3716–3730.

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

Wu, M., Xuan, J., and Lu, J. (2023). Indirect functional bayesian neural networks. In *Fifth Symposium on Advances in Approximate Bayesian Inference*.

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

Yin, L. and Ao, P. (2006). Existence and construction of dynamical potential in nonequilibrium processes without detailed balance. *Journal of Physics A: Mathematical and General*, 39(27):8593.

Zhang, C., Bütepage, J., Kjellström, H., and Mandt, S. (2018). Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):2008–2026.

Zhang, R., Li, C., Zhang, J., Chen, C., and Wilson, A. G. (2020). Cyclical stochastic gradient mcmc for bayesian deep learning. In *International Conference on Learning Representations*.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets. [Yes]

(b) The license information of the assets, if applicable. [Not Applicable]

(c) New assets either in the supplemental material or as a URL, if applicable. [Yes]

(d) Information about consent from data providers/curators. [Not Applicable]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A NOTATION TABLE

Table 5 is the notation table to demonstrate the notation used in this paper.

Table 5: Notation table

Notation	Meanings
$\mathcal{D} = \{\mathbf{X}_\mathcal{D}, \mathbf{Y}_\mathcal{D}\}$	Training dataset
$\mathcal{X} \subseteq \mathbb{R}^p$	(p-dimensional) input space
$\mathcal{Y} \subseteq \mathbb{R}^c$	(c-dimensional) output space
$(\Omega, \mathcal{A}, P)$	Probability space on $\mathbb{R}^k$
$\mathbb{H}$	Infinite-dimensional function space (Banach space)
$\mathcal{B}(\mathbb{H})$	Borel σ-algebra on $\mathbb{H}$
$\mathcal{P}(\mathbb{H})$	The space of Borel probability measures on $\mathcal{B}(\mathbb{H})$.
X	Stochastic process
$\mathbf{X}_\mathcal{M}$	Finite measurement points from input space
$\mathbf{w} \in \mathbb{R}^k$	Random network parameters
$\mathbf{z} \in \mathbb{R}^k$	Momentum variable
ϵ	Discretization step size
η	Standard Gaussian noise
$f(\cdot; \mathbf{w})$	Random function mapping defined by a BNN parameterized by $\mathbf{w}$
$p_0(\mathbf{w})$	Prior distribution over network parameters
$p(\mathbf{w}\|\mathcal{D})$	Posterior distribution over network parameters
$p(\mathbf{Y}_\mathcal{D}\|f(\mathbf{X}_\mathcal{D}; \mathbf{w}))$	Likelihood function evaluated on the training data
$p(\mathbf{z})$	Auxiliary probability distribution
P_0	Functional Prior measure
$\text{supp}(P_0)$	Topological support of P_0
E	A non-empty subset of $\text{supp}(P_0)$
$P_{f\|\mathcal{D}}$	Posterior measure over functions
$\Phi(f)$	Negative log-likelihood as $-\log p(\mathbf{Y}_\mathcal{D}\|f(\mathbf{X}_\mathcal{D}; \mathbf{w}))$
$p(g)$	Functional auxiliary probability measure
$U(\mathbf{w})$	Potential energy function in parameter-space Langevin dynamics
$H(\mathbf{w}, \mathbf{z})$	Potential energy function in parameter-space Hamiltonian dynamics
$I_0(f)$	Onsager–Machlup (OM) functional for P_0
$U(f)$	Potential energy functional in functional Langevin dynamics
$H(f, g)$	Potential energy functional in functional Hamiltonian dynamics
$\mu(\cdot)$	Drift term in the SDE of Itô diffusion
$\sigma(\cdot)$	Diffusion coefficient in the SDE of Itô diffusion
B	Standard Wiener process (Brownian motion)
$\pi(\cdot)$	Stationary distribution (measure)

B FURTHER BACKGROUND

Itô Diffusion Dynamics-based MCMC methods are rooted in the general framework of Itô diffusion (Øksendal, 2003), which is commonly used to model the evolution of particles in a system. Given a stochastic process $X : [0, \infty) \times \Omega \to \mathbb{R}^n$ defined on a probability space (Ω, Σ, P), an Itô diffusion in n-dimensional Euclidean space, driven by the standard Wiener process (Brownian motion), satisfies the following specific type of stochastic differential equation (SDE):

$$\mathrm{d}X_t = \mu(X_t)\mathrm{d}t + \sigma(X_t)\mathrm{d}B_t, \tag{11}$$

where X_t represents the state of the stochastic process at time t, $\mu(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ is a vector field describing the deterministic drift term, and $\sigma(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is a matrix field denotes the diffusion coefficient for X_t. Both $\mu(\cdot)$ and $\sigma(\cdot)$ are assumed to satisfy the standard Lipschitz continuity condition (Ghosh, 2010). The term $B \in \mathbb{R}^n$ refers to an n-dimensional Wiener process (Brownian motion), with $\mathrm{d}B_t$ being the increment of a Wiener process, distributed as $\mathrm{d}B_t \sim \mathcal{N}(0, \mathrm{d}t \cdot \mathbf{I}_n)$. The probability density function $p(x, t)$ of X_t is governed by the Fokker-Planck (FP) equation (Risken, 1996):

$$\frac{\partial p(x,t)}{\partial t} = -\sum_i \frac{\partial}{\partial x_i}\left[\mu_i(X_t)p(x,t)\right] + \sum_{i,j} \frac{\partial^2}{\partial x_i \partial x_j}\left[D_{i,j}(x,t)p(x,t)\right], \tag{12}$$

where $D_{i,j}(x, t) = \frac{1}{2}\sigma(X_t)\sigma(X_t)^T$. This equation serves as the foundation for deriving dynamics-based MCMC sampling methods. The diffusion process reaches a stationary state when the FP equation equals zero, then

simulating this diffusion process is equivalent to sampling from the stationary distribution, denoted by $\pi(x)$. By carefully designing $\mu(\cdot)$ and $\sigma(\cdot)$, one can sample from the target distribution as the stationary distribution. For example, for the commonly used Langevin dynamics, where $\mu(X) = -\nabla U(X)$, $\sigma(X) = \sqrt{2} \cdot \mathbf{I}_n$, and $U(\cdot)$ is a potential energy function. The stationary distribution for $p(x,t)$ is given by the Boltzmann distribution $\pi(x) \propto \exp(-U(X))$. Thus, when sampling the posterior $p(\mathbf{w}|\mathcal{D})$ in BNNs, the potential energy function is given by $U(\mathbf{w}) = -\log p(\mathbf{Y}_\mathcal{D}|\mathbf{X}_\mathcal{D}; \mathbf{w}) - \log p_0(\mathbf{w})$, and $\sigma(\mathbf{w}) = \sqrt{2} \cdot \mathbf{I}_k$, ensuring that the target posterior as the stationary distribution.

Itô Lemma Itô Lemma (Itô, 1951) is a fundamental result in stochastic calculus to find the differential of a function of a stochastic process, which serves as the stochastic calculus counterpart of the chain rule. For an Itô diffusion, let $f(X_t)$ be an arbitrary twice differentiable scalar function of real variables X_t, then the differential of $f(X_t)$ can be derived from the Taylor series expansion of the function as

$$\mathrm{d}f(X_t) = \left(\mu(X_t)\frac{\partial f}{\partial x} + \frac{\sigma^2(X_t)}{2}\frac{\partial^2 f}{\partial x^2}\right)\mathrm{d}t + \sigma(X_t)\frac{\partial f}{\partial x}\mathrm{d}B_t, \tag{13}$$

which implies that $f(X_t)$ is itself an Itô diffusion (Brzeźniak et al., 2008).

Onsager–Machlup Functional (see e.g., Lambley (2023), Definition 2.4.) Suppose $\mathbb{H}$ is a separable metric space and $P_0 \in \mathcal{P}(\mathbb{H})$. Assume that $\varnothing \neq E \subseteq \mathrm{supp}(P_0)$, then a functional $I_0 : E \to \mathbb{R}$ is an Onsager–Machlup (OM) functional for P_0 if for all $f, f' \in E$,

$$\lim_{r\to 0} \frac{P_0(B_r(f)}{P_0(B_r(f')} = \exp(I_0(f') - I_0(f)), \tag{14}$$

where $B_r(\cdot)$ denotes the closed ball of radius r in $\mathbb{H}$ and $\mathrm{supp}(P_0) := \{f \in \mathbb{H} \mid P_0(B_r(f)) > 0 \text{ for all } r > 0\}$. Lambley (2023) (in Proposition 2.6.) proved that posterior measure $P_{f|\mathcal{D}} \in \mathcal{P}(\mathbb{H})$ defined as $P_{f|\mathcal{D}}(\mathrm{d}f) \propto \exp(-\Phi(f))P_0(\mathrm{d}f)$ has OM functional $I^y : E \to \mathbb{R}$ given by $I^y(f) = I_0(f) + \Phi(f)$, where $\Phi(f)$ is the continuous *potential*, in essence the negative log-likelihood as $-\log p(\mathbf{Y}_\mathcal{D}|f(\mathbf{X}_\mathcal{D}; \mathbf{w}))$. OM functional can be interpreted heuristically as the negative logarithm of the Lebesgue density, but this interpretation cannot be taken literally in infinite-dimensional spaces, as there is no Lebesgue measure in such settings. For instance, the OM functional for a Gaussian measure on an infinite-dimensional Banach space is typically defined only on a specific subspace, known as the Cameron-Martin space, e.g., for a centred Gaussian measure, the OM functional is defined as $I : E \to \mathbb{R}, I(f) = \frac{1}{2}||f||_E^2$. As a result, the OM functional does not need to be defined over the entire space $\mathbb{H}$.

C THEORETICAL PROOF

The proof of Proposition 3.1 and Proposition 3.2 will be concise based on the general framework for dynamics-based sampling methods proposed by Ma et al. (2015). They proved in Theorem 1 that if $\mu(X_t)$ and $\sigma(X_t)$ are restricted to the following form

$$\begin{aligned}
\mu(X) =& -[\mathbf{D}(X) + \mathbf{Q}(X)]\nabla G(X) + \Gamma(X),\\
\Gamma(X) =& \sum \frac{\partial}{\partial X}(\mathbf{D}_{ij}(X) + \mathbf{Q}_{ij}(X)),\\
\sigma(X) =& \sqrt{2\mathbf{D}(X)},
\end{aligned} \tag{15}$$

where $G(\cdot)$ is the potential energy functional, $\mathbf{D}(\cdot)$ is a positive semidefinite matrix, and $\mathbf{Q}(\cdot)$ is a skew-symmetric curl matrix, then the Fokker–Planck equation can be transformed into a more compact form (Yin and Ao, 2006; Shi et al., 2012) given by

$$\frac{\partial p(x,t)}{\partial t} = \nabla^T \cdot ([\mathbf{D}(X) + \mathbf{Q}(X)][p(x,t)\nabla G(X) + \nabla p(x,t)]). \tag{16}$$

Then, it is straightforward to verify that the stationary distribution of the diffusion process is exactly as $\pi(X) \propto \exp(-G(X))$.

For the proof of Proposition 3.1, it is clear that the functional Langevin dynamics in Equation (3) can be cast into the above general framework as

$$\begin{aligned}
\mathrm{d}f_t(\cdot; \mathbf{w}) &= [-(\nabla_\mathbf{w} f_t)^T(\nabla_\mathbf{w} f_t)\nabla_f U(f_t) + H_\mathbf{w} f]\mathrm{d}t + \sqrt{2}(\nabla_\mathbf{w} f_t)^T \mathrm{d}B_t\\
&= (-[\mathbf{D}(f) + \mathbf{Q}(f)]\nabla_f U(f_t) + \Gamma(f))\,\mathrm{d}t + \sqrt{2\mathbf{D}(f)}\mathrm{d}B_t,
\end{aligned} \tag{17}$$

Table 6: General framework of naive SGMCMC and functional SGMCMC algorithms.

Dynamics	X	$G(X)$	$\mathbf{Q}(X)$	$\mathbf{D}(X)$
SGLD	$\mathbf{w}$	$\tilde{U}(\mathbf{w})$	$\mathbf{0}$	$\mathbf{I}_k$
fSGLD	f	$\tilde{U}(f)$	$\mathbf{0}$	$(\nabla_{\mathbf{w}} f)^T(\nabla_{\mathbf{w}} f)$
SGHMC	$(\mathbf{w}, \mathbf{z})$	$\tilde{H}(\mathbf{w}, \mathbf{z})$	$\begin{pmatrix} \mathbf{0} & -\mathbf{I}_k \\ \mathbf{I}_k & \mathbf{0} \end{pmatrix}$	$\begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & C \end{pmatrix}$
fSGHMC	(f, g)	$\tilde{H}(f, g)$	$\begin{pmatrix} \mathbf{0} & -(\nabla_{\mathbf{w}} f)^T \nabla_{\mathbf{z}} g \\ (\nabla_{\mathbf{z}} g)^T \nabla_{\mathbf{w}} f & \mathbf{0} \end{pmatrix}$	$\begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & C(\nabla_{\mathbf{z}} g)^T \nabla_{\mathbf{z}} g \end{pmatrix}$

where $U(f) = -\log p(\mathbf{Y}_{\mathcal{D}} | f(\mathbf{X}_{\mathcal{D}}; \mathbf{w})) + I_0(f)$ is an OM functional for $P_{f|\mathcal{D}}$, $\mathbf{D}(f) = (\nabla_{\mathbf{w}} f)^T(\nabla_{\mathbf{w}} f)$, $\mathbf{Q}(f) = \mathbf{0}$, $\Gamma(f) = H_{\mathbf{w}} f$. Therefore, the Fokker-Planck equation of the probability measure $p(f, t)$ for $f_t(\cdot; \mathbf{w})$ then can be derived as

$$\frac{\partial p(f,t)}{\partial t} = \nabla^T \cdot ([\mathbf{D}(f) + \mathbf{Q}(f)][p(f,t)\nabla_f U(f) + \nabla p(f,t)]) \,, \tag{18}$$

from which the stationary probability measure of $p(f, t)$ can be verified as $\exp(-U(f)) = P_{f|\mathcal{D}}$, that is, the target posterior measure over functions.

For the proof of Proposition 3.2, the functional Hamiltonian dynamics defined in Equation (7) can be written as

$$\begin{aligned} \mathrm{d} \begin{bmatrix} f_t(\cdot; \mathbf{w}) \\ g_t(\cdot; \mathbf{z}) \end{bmatrix} &= - \begin{bmatrix} 0 & -(\nabla_{\mathbf{w}} f)^T \nabla_{\mathbf{z}} g_t \\ (\nabla_{\mathbf{z}} g)^T \nabla_{\mathbf{w}} f_t & 0 \end{bmatrix} \begin{bmatrix} \nabla_f U(f_t) \\ -\nabla_g \log p(g_t) \end{bmatrix} \mathrm{d}t \\ &= -\mathbf{Q}(f, g) \nabla H(f_t, g_t) \mathrm{d}t, \end{aligned} \tag{19}$$

where $\mathbf{Q}(f, g)$ is s a skew-symmetric curl matrix, $\mathbf{D}(f, g) = \mathbf{0}$. Therefore, the stationary measures of the functional Hamiltonian dynamics defined in Equation (7) is $\pi(f, g) \propto \exp(H(f, g))$, and simply (marginalize) discard the auxiliary functions can obtain the target true posterior $P_{f|\mathcal{D}}$.

For the stochastic gradient version, the functional stochastic gradient Hamiltonian dynamics (fSGHMC) for the discretization update rule for samples of $\mathbf{w}$ and $\mathbf{z}$ defined in Equation (9) is defined as follows:

$$\mathrm{d} \begin{bmatrix} f_t(\cdot; \mathbf{w}) \\ g_t(\cdot; \mathbf{z}) \end{bmatrix} = - \begin{bmatrix} 0 & -(\nabla_{\mathbf{w}} f)^T \nabla_{\mathbf{z}} g_t \\ (\nabla_{\mathbf{z}} g)^T \nabla_{\mathbf{w}} f_t & C(\nabla_{\mathbf{z}} g)^T \nabla_{\mathbf{z}} g_t \end{bmatrix} \begin{bmatrix} \nabla_f \tilde{U}(f_t) \\ -\nabla_g \log p(g_t) \end{bmatrix} \mathrm{d}t + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sqrt{2C}(\nabla_{\mathbf{z}} g_t)^T \end{bmatrix} \mathrm{d}t, \tag{20}$$

where $\mathbf{D}(f, g) = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & C(\nabla_{\mathbf{z}} g)^T \nabla_{\mathbf{z}} g \end{pmatrix}$ and $\mathbf{Q}(f, g) = \begin{pmatrix} \mathbf{0} & -(\nabla_{\mathbf{w}} f)^T \nabla_{\mathbf{z}} g \\ (\nabla_{\mathbf{z}} g)^T \nabla_{\mathbf{w}} f & \mathbf{0} \end{pmatrix}$ is the same as that in full-batch functional Hamiltonian dynamics. Then, the (marginal) stationary probability measure of such functional stochastic gradient Hamiltonian is still the target functional posterior $P_{f|\mathcal{D}}$.

Moreover, the comparisons of matrix $\mathbf{D}(\cdot)$ and $\mathbf{Q}(\cdot)$ for naive SGMCMC methods and our functional SGMCMC are shown in Table 6.

D PSEUDOCODE FOR FUNCTIONAL SGMCMC.

Algorithm 1 presents the pseudocode for fSGLD. And the pseudocode for fSGHMC is shown in Algorithm 2.

E EXPERIMENTAL SETTING

Extrapolation on Synthetic Data In this experiment, we use 2×100 fully connected tanh neural networks for all models. The functional GP prior with the RBF kernel is pre-trained on the 20 training points for 100 epochs for all functional methods. we use a decreasing step-size schedule for ϵ_t with 1e-3 as the initial value for all sampling methods. The specific decreasing schedule is as every 5000 iterations, the decay factor is 0.9.

Multivariate Regression on UCI Datasets In this experiment, we use two-hidden-layer fully connected neural networks and each layer with 10 hidden units for all models. The functional GP prior with the RBF kernel

Algorithm 1 Functional SGLD (fSGLD)

Input: Dataset $\mathcal{D}$, pre-trained GP prior $p(f)$, initialized $\mathbf{w}_0 \sim \mathcal{N}(0, I)$, step size ϵ_t, noise $\eta_t \sim \mathcal{N}(0, 1)$, number of burn-in iterations N_b, number of sample K
# Burn-in stage :
for $t = 0$ **to** N_b **do**
  draw measurement set $\mathbf{X}_{\mathcal{M}}$;
  update $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \epsilon_t \nabla_{\mathbf{w}} \tilde{U}(f_t)) + \sqrt{2\epsilon_t}\eta_t$ using Equation (6);
end for
# Sampling stage :
$S \leftarrow \varnothing$;
$\mathbf{w}_0 \leftarrow \mathbf{w}_{N_b}$;
for $t = 0$ **to** K **do**
  draw measurement set $\mathbf{X}_{\mathcal{M}}$;
  $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \epsilon_t \nabla_{\mathbf{w}} \tilde{U}(f_t)) + \sqrt{2\epsilon_t}\eta_t$ using Equation (6);
  $S \leftarrow S \cup \{\mathbf{w}_{t+1}\}$;
end for

Algorithm 2 Functional SGHMC (fSGHMC)

Input: Dataset $\mathcal{D}$, pre-trained GP prior $p(f)$, initialized $\mathbf{w}_0 \sim \mathcal{N}(0, I)$, $\mathbf{z}_0 \sim \mathcal{N}(0, M)$, step size ϵ_t, noise $\eta_t \sim \mathcal{N}(0, 1)$, number of burn-in iterations N_b, leapfrog steps m, number of sample K
# Burn-in stage :
for $t = 0$ **to** N_b **do**
  resample momentum $\mathbf{z}_t \sim \mathcal{N}(0, M)$
  $(\mathbf{w}_0, \mathbf{z}_0) \leftarrow (\mathbf{w}_t, \mathbf{z}_t)$;
  for $i = 1$ **to** m **do**
    draw measurement set $\mathbf{X}_{\mathcal{M}}$;
    update $(\mathbf{w}_i, \mathbf{z}_i)$ using Equation (9);
  end for
  $(\mathbf{w}_{t+1}, \mathbf{z}_{t+1}) \leftarrow (\mathbf{w}_m, \mathbf{z}_m)$
end for
# Sampling stage :
$S \leftarrow \varnothing$;
$\mathbf{w}_0 \leftarrow \mathbf{w}_{N_b}$;
for $t = 0$ **to** K **do**
  resample momentum $\mathbf{z}_t \sim \mathcal{N}(0, M)$
  $(\mathbf{w}_0, \mathbf{z}_0) \leftarrow (\mathbf{w}_t, \mathbf{z}_t)$;
  for $i = 1$ **to** m **do**
    draw measurement set $\mathbf{X}_{\mathcal{M}}$;
    update $(\mathbf{w}_i, \mathbf{z}_i)$ using Equation (9);
  end for
  $(\mathbf{w}_{t+1}, \mathbf{z}_{t+1}) \leftarrow (\mathbf{w}_m, \mathbf{z}_m)$
  $S \leftarrow S \cup \{\mathbf{w}_{t+1}\}$;
end for

Table 7: Runtime comparison of single iteration for all methods (in seconds).

Model	1-D extrapolation	UCI	Image classification
SGLD	0.0016	0.0025	4
fSGLD	0.0044	0.0045	5.83
SGHMC	0.0023	0.02	10.95
fSGHMC	0.0051	0.044	13.64
IFBNN	0.0094	0.0075	6.81
FBNN	0.129	0.114	23.25

is pre-trained for 100 epochs. We run 500 iterations for the burn-in stage and collect 15 samples in the following 1500 iterations for all sampling methods. The initial ϵ_t is 1e-3, and the decay factor is 0.9. Functional variational inference methods are trained for 2000 epochs for fair comparison.

Imagine Classification and OOD Detection In this experiment, we use ResNet-18 architecture for all methods. The functional prior is a Dirichlet-based GP designed for classification tasks and is pre-trained for 500 epochs. The initial ϵ_t is 1e-2, the decay period is 20, and the decay factor is 0.99. For all sampling methods, we run for 60 burn-in iterations and collect 10 samples in the following 20 iterations on the MNIST dataset, 100 burn-in iterations and collect 10 samples in the following 100 iterations on Fashion-MNIST, 80 burn-in iterations and draw 10 samples in the following 20 iterations on CIFAR10. Functional variational inference methods are trained for 80, 200, and 100 epochs on MNIST, FashionMNIST and CIFAR-10, respectively.

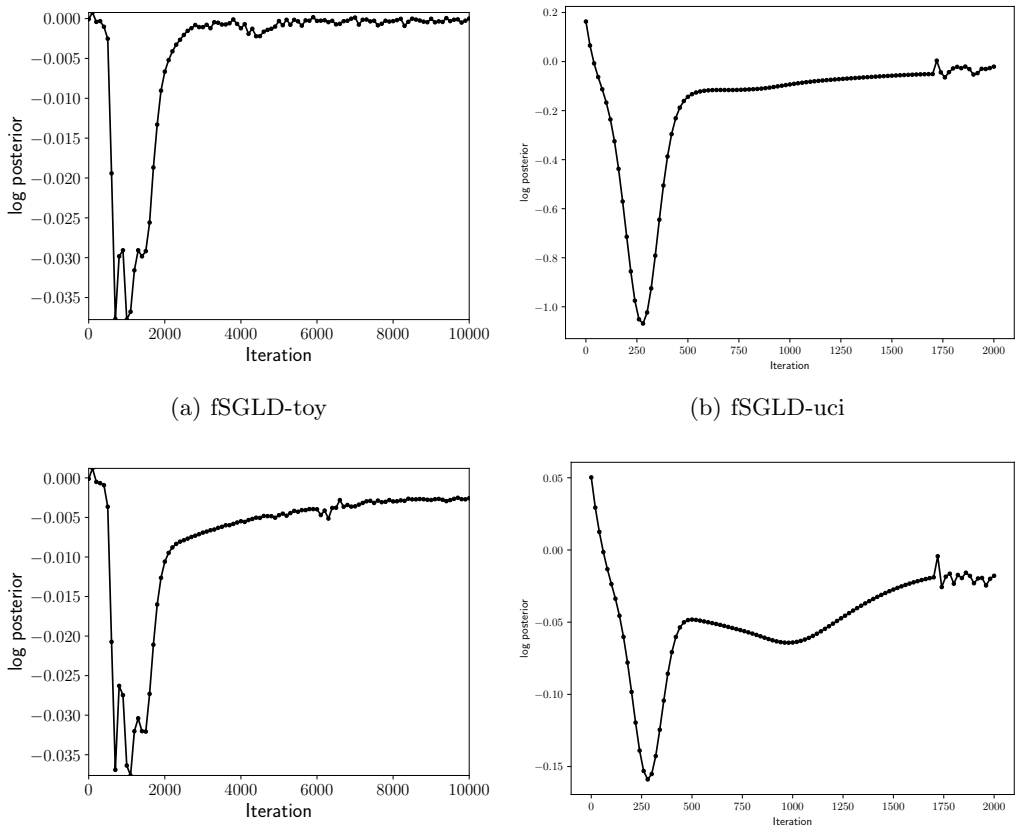
Contextual Bandits In this experiment, we use fully connected neural networks with input-100-100-output architecture for all methods. The GP prior with RBF kernel is pre-trained on 1000 randomly sampled points from training data for 100 epochs. The initial ϵ_t is 1e-2, and the decay factor is 0.9. All models are trained using the last 4096 input-output tuples in the training buffer with a batch size of 64 and training frequency of 64 for each iteration.

F ANALYSIS OF MIXING TIME

The trajectories of our fSGLD and the naive parameter-space SGLD for both the toy synthetic extrapolation and UCI regression (Yacht) experiments are shown in Figure 3. For the extrapolation example, we use a two-hidden-layer fully connected neural network with 100 hidden units in each layer (resulting in a 10401-dimensional parameter space). For the UCI regression task, we use a 2×10 fully connected network (141-dimensional parameter space). As illustrated in Figure 3, our fSGLD converges rapidly to the stationary measure—within 2000 iterations for the extrapolation example and 500 iterations for the UCI regression task. This convergence rate is comparable to that of the parameter-space SGLD, highlighting the rapid mixing rate of our method.

G COMPUTATIONAL COMPLEXITY

The runtime comparisons of a single iteration for all methods on the synthetic extrapolation, UCI regression (yacht) and image classification (MNIST) are provided in Table 7. For our fSGLD, fSGHMC, functional variational IFBNN and FBNN methods, the functional GP prior was pre-trained for 100, 100, and 500 epochs, respectively, in these three experiments, which added only an extra 1s for both the 1-D and UCI experiments, and 3s for the classification task. Compared to functional variational inference methods, our functional SGMCMC schemes exhibit significantly better computational performance, and are only slightly slower than the naive parameter-space SGMCMC. This demonstrates the computational efficiency of our method while maintaining the advantages of incorporating functional priors for improved inference.



(a) fSGLD-toy (b) fSGLD-uci

(c) SGLD-toy (d) SGLD-uci

Figure 3: Log-posterior probability versus the number of iterations.

Table 8: The table shows the average RMSE results for sample size effects (150 samples).

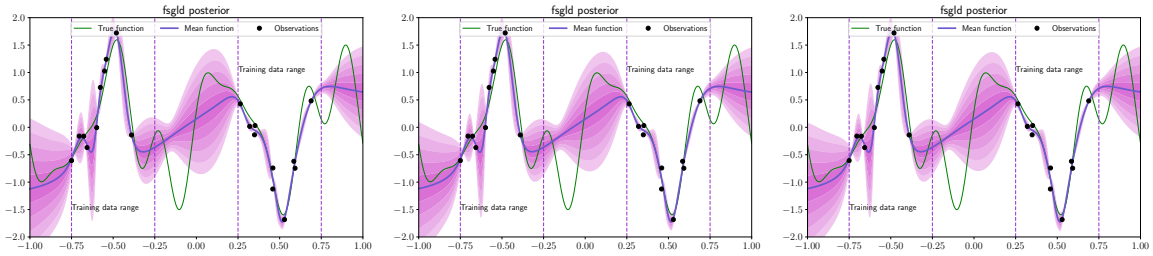
	RMSE			
	SGLD	fSGLD	SGHMC	fSGHMC
Yacht	1.095 ± 0.128	0.409 ± 0.112	1.187 ± 0.141	**0.246 ± 0.142**
Boston	1.248 ± 0.072	0.392 ± 0.112	1.332 ± 0.068	**0.270 ± 0.135**

H ABLATION STUDY

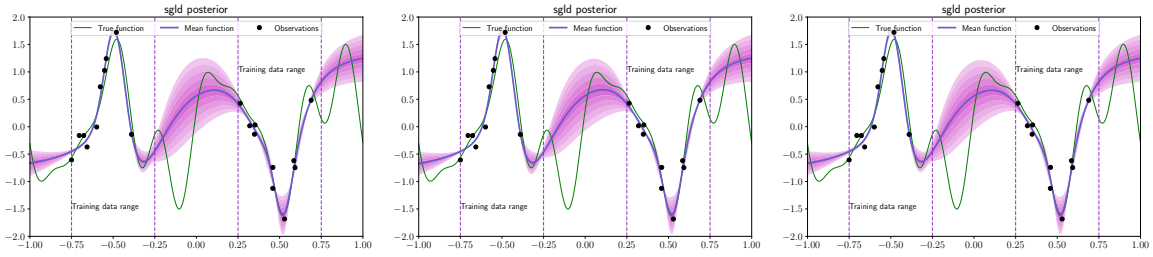
H.1 The Effects of the Sample Size

In this section, we investigate the effect of sample size on the results of naive SGMCMC and our functional SGMCMC by varying the number of samples. For the 1-D extrapolation example in Section 5.1. We consider three different sample sizes for SGLD, SGHMC, fSGLD, and fSGHMC: 10, 80, and 200, respectively. The extrapolation results are shown in Figure 4. We can see that no matter the sample size, there is almost no difference in the fitting effect and uncertainty estimation for both naive SGMCMC and our functional SGMCMC, which indicates that the sample size has almost no effect on our experimental results.

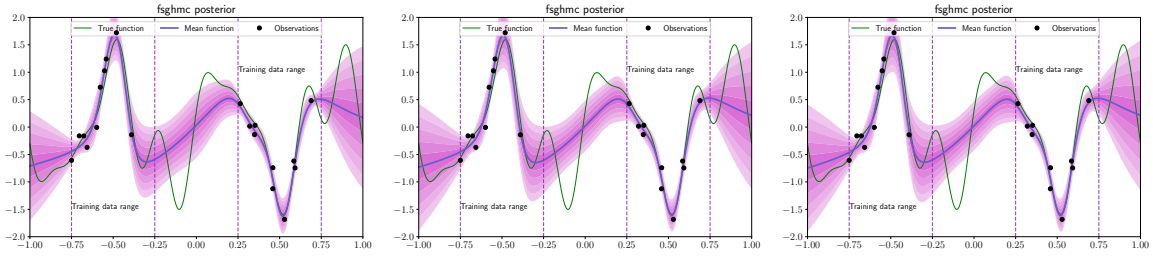
For the UCI regressions in Section 5.2, the sample size is 15 for all naive SGMCMC and functional SGMCMC methods. We now consider a bigger sample size of 150 for all methods on the *Yacht* and *Boston* datasets. The RMSE and NLL results for the naive SGMCMC and functional SGMCMC are shown in Table 8 and Table 9, respectively. We randomly split each dataset into 90% training data and 10% test data, which is repeated 5 times to ensure validity. These two results have only very slight fluctuations compared to the results in Table 2 and Table 3, which still illustrates that our results are almost independent of the sample size. Our functional fSGLD and fSGHMC consistently demonstrate superior performance.



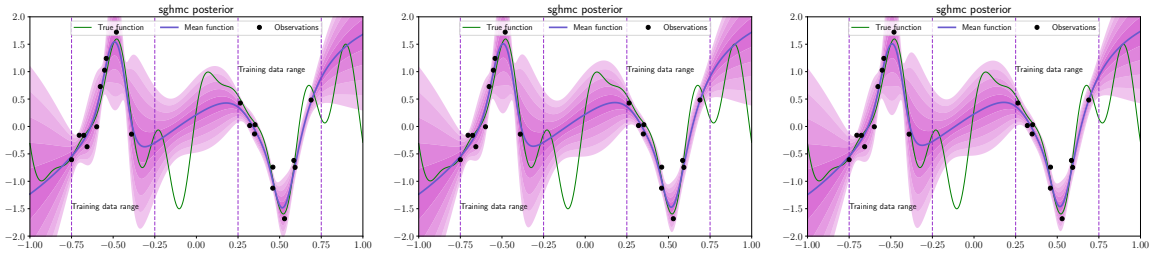
(a) fSGLD-10 (b) fSGLD-80 (c) fSGLD-200



(d) SGLD-10 (e) SGLD-80 (f) SGLD-200



(g) fSGHMC-10 (h) fSGHMC-80 (i) fSGHMC-200

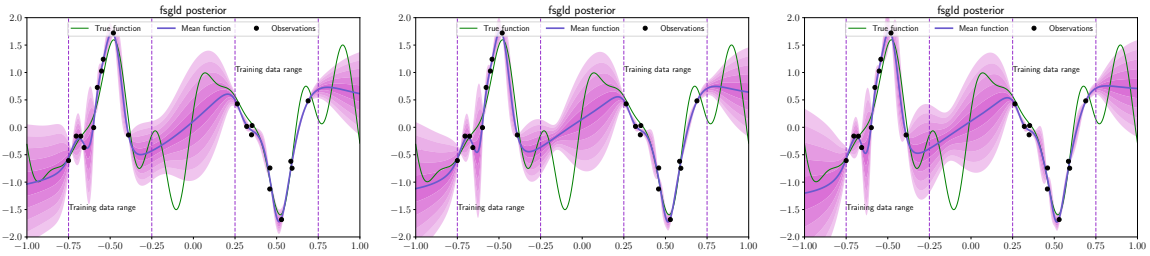


(j) SGHMC-10 (k) SGHMC-80 (l) SGHMC-200

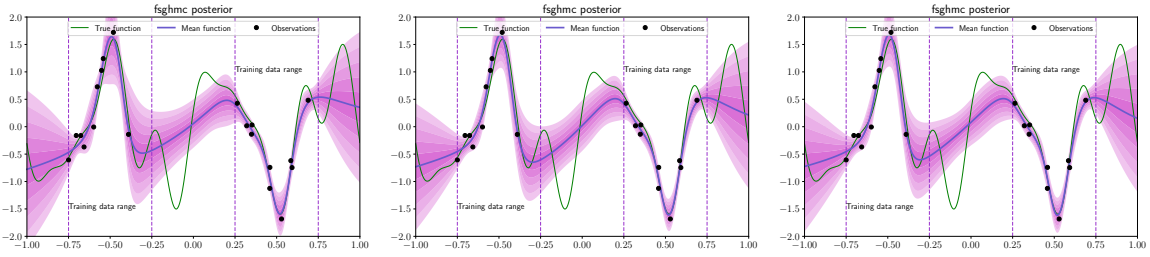
Figure 4: The effect of sample size on 1-D extrapolation example for fSGLD, SGLD, fSGHMC, SGHMC. The number after the short line in each subheading represents the sample size.

Table 9: The table shows the average NLL results for sample size effects (150 samples).

	NLL			
	SGLD	fSGLD	SGHMC	fSGHMC
Yacht	-0.352 ± 0.073	-2.332 ± 0.493	-2.300 ± 0.181	**-3.406 ± 0.517**
Boston	-0.509 ± 0.129	-2.068 ± 0.159	-2.352 ± 0.100	**-3.060 ± 0.284**



(a) fSGLD-m10 (b) fSGLD-m40 (c) fSGLD-m400



(d) fSGHMC-m10 (e) fSGHMC-m40 (f) fSGHMC-m400

Figure 5: The effects of the number of measurement points on the gradient estimation of functional prior for fSGLD and fSGHMC. The number after the short line in each subheading represents the number of measurement points.

H.2 The Effects of the Number of Measurement Points on the Gradient Estimation of Functional Prior

In the 1-D extrapolation example in Section 5.1, we sampled 40 measurement points from 20 training data and an additional 40 inducing points from Uniform $(-1, 1)$ to estimate the gradient of log functional prior for fSGLD and fSGHMC. We now investigate the effects of the number of measurement points on the approximate estimation of the gradient of the log functional prior in functional SGMCMC. We consider two other cases for the number of measurement points: 10 measurement points sampled from 20 training data and an additional 40 inducing points from Uniform $(-1, 1)$; and 400 measurement points sampled from 20 training data and an additional 1000 inducing points from Uniform $(-1, 1)$. The comparison results are shown in Figure 5. We can see that there is almost no difference in the prediction performance for the three different numbers of measurement points. The uncertainty estimations also fluctuate only slightly. Overall, we can conclude that the number of measurement points for gradient estimation of functional prior does not significantly impact the results of our methods on 1-D extrapolation example.

In the UCI regressions in Section 5.2, for datasets with sample sizes less than 1000, we estimate the gradient of the functional prior for fSGLD and fSGHMC using the training data as the finite measurement points. For datasets with sample sizes larger than 1000, we randomly sample 1000 data from the training data as measurement points. We now consider using only 40 measurement points on the *Yacht* and *Boston* datasets. The results of RMSE and NLL are shown in Table 10 and Table 11, respectively. Compared to the results in Table 2 and Table 3, the RMSE results are slightly worse for both fSGLD and fSGHMC. The NLL results for fSGLD get slightly worse on the *Yacht* dataset and remain essentially unchanged on the *Boston* dataset. The NLL results for fSGHMC are slightly worse on both datasets but still significantly outperforms all other methods. Overall, for the UCI regressions, the more measurement points we used, the better the performance of our methods, probably because more measurement points are more accurate for the gradient estimation of the functional prior.

H.3 The Effects of the Functional GP Prior

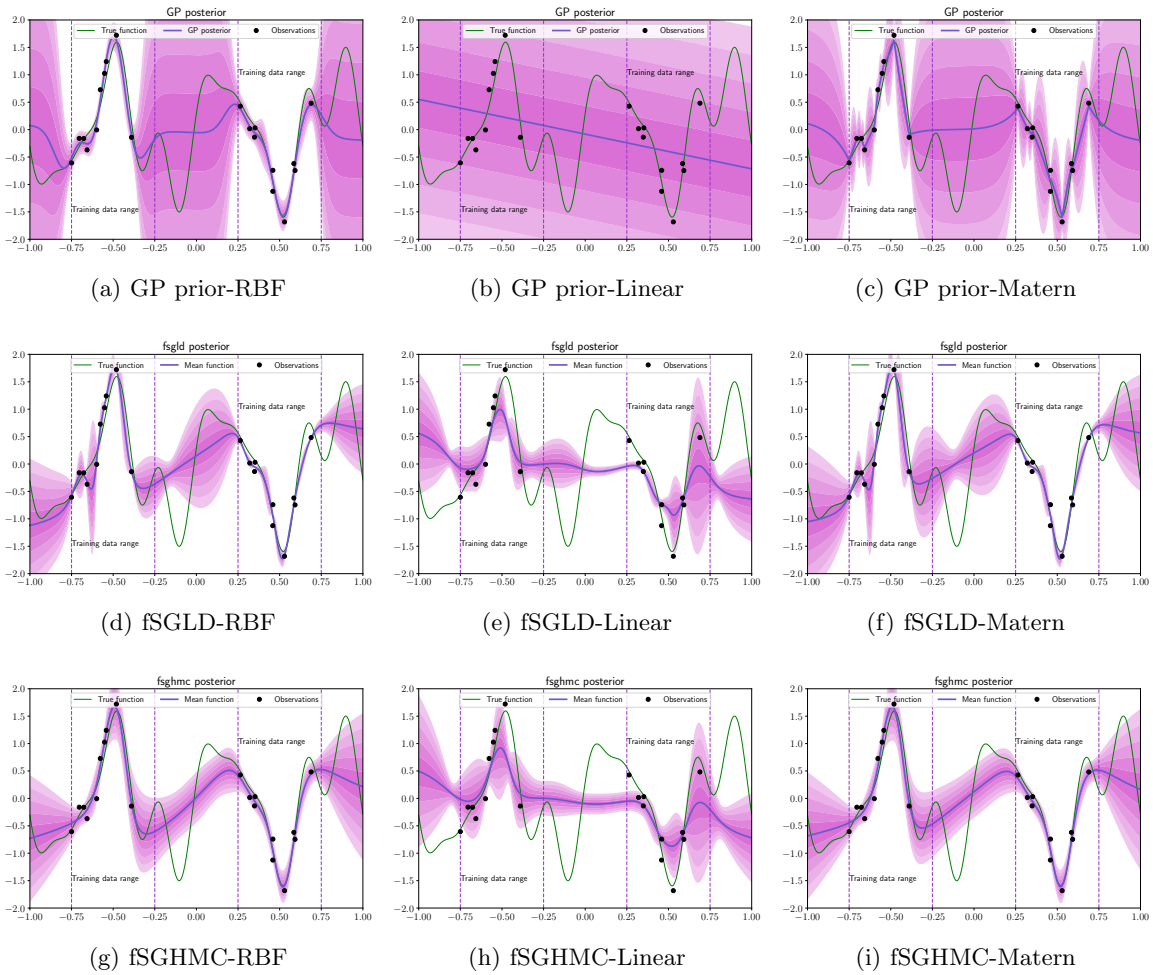
Functional Gaussian processes (GPs) priors are able to encode prior knowledge about function properties (e.g., periodicity and smoothness) through corresponding kernel functions. We used RBF kernel in the 1-D extrapolation

Table 10: The table shows the average RMSE results for the effects of the number of measurement points.

	RMSE	
	fSGLD	fSGHMC
Yacht	0.496 ± 0.072	0.297 ± 0.111
Boston	0.448 ± 0.098	0.290 ± 0.126

Table 11: The table shows the average NLL results for the effects of the number of measurement points.

	NLL	
	fSGLD	fSGHMC
Yacht	-2.327 ± 0.564	-2.738 ± 0.428
Boston	-2.277 ± 0.089	-2.607 ± 0.329



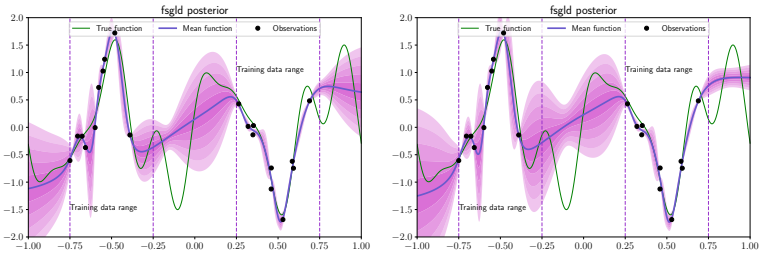
(a) GP prior-RBF (b) GP prior-Linear (c) GP prior-Matern

(d) fSGLD-RBF (e) fSGLD-Linear (f) fSGLD-Matern

(g) fSGHMC-RBF (h) fSGHMC-Linear (i) fSGHMC-Matern

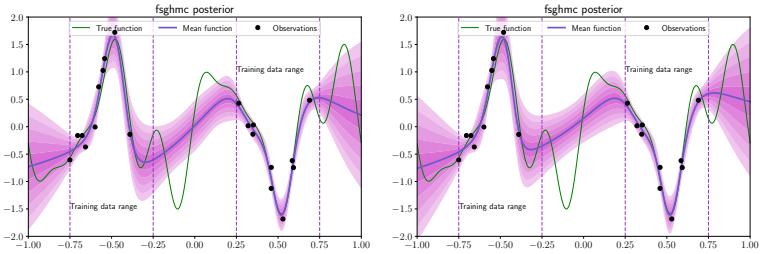
Figure 6: The effects of the kernel function on the functional GP prior. The text after the short line in each subheading represents the corresponding kernel functions.

experiment, which is suited for modelling polynomial functions. We now consider two other kernel functions of GP priors to our functional SGMCMC: Matern kernel and Linear kernel (unsuitable for modelling polynomial oscillatory curves). The results are shown in Figure 6. The top row presents the GP prior predictions for RBF kernel, Linear kernel and Matern kernel, respectively. The results from the mismatched Linear kernel in Figure 6(e) and Figure 6(h) for fSGLD and fSGHMC, respectively, show that the fitting performance deteriorates in the observation region and also demonstrate a strong linear trend (converging to a horizontal line) in the middle unseen region $[-0.25, 0.25]$ since the Linear GP prior prediction shows a completely linear trend and falls to fit



(a) fSGLD

(b) fSGLD



(c) fSGHMC

(d) fSGHMC

Figure 7: 1-D extrapolation example without prior pre-training. (a) and (c) are the results with pre-trained prior; (b) and (d) are results without pre-trained prior.

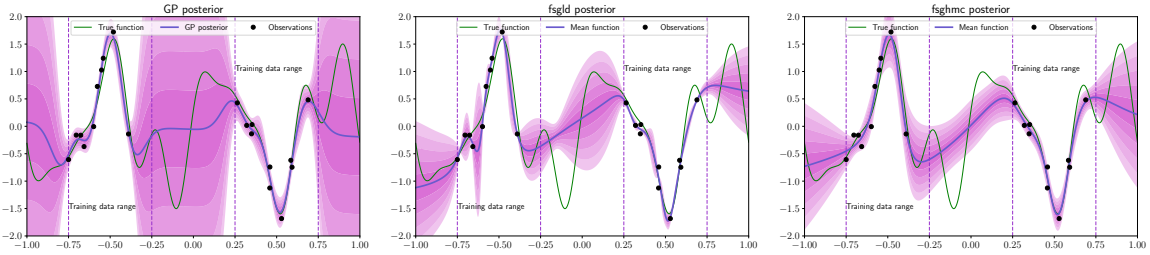
Table 12: RMSE and NLL of fSGLD and fSGHMC on UCI regression (yacht) with no pre-trained GP priors.

Model	pre-trained GP	fixed GP
RMSE		
fSGLD	0.41	0.58
fSGHMC	0.25	0.51
NLL		
fSGLD	-2.46	-1.65
fSGHMC	-3.37	-2.52

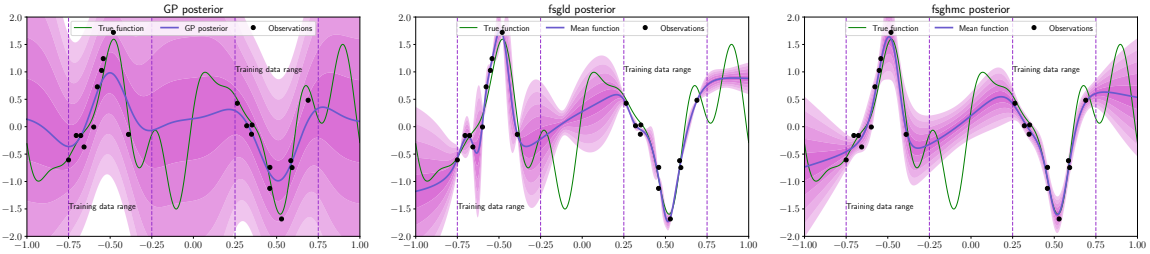
the target function. The uncertainty intervals in the leftmost $[-1, -0.75]$ and rightmost $[0.75, 1]$ non-observation regions deviate far from the true function. On the contrary, the results of the Matern kernel are hardly different from those of the RBF kernel due to the fact that it also has a strong ability to describe polynomial functions. This indicates that our method can effectively incorporate functional prior information into the posterior inference.

We also plot the predictions of fSGLD and fSGHMC with no pre-trained GP priors in Figure 7. The leftmost columns, Figure 7(a) and Figure 7(c), show the results for fSGLD and fSGHMC with pre-trained GP priors, as described in the main paper. The rightmost columns, Figure 7(b) and Figure 7(d), display the predictions without pre-trained GP priors, using a fixed constant mean function $m(x) = 0$ and a kernel function $k(x, x') = 1$. Despite not having pre-trained priors, the posterior results remain quite similar to those with pre-trained GP priors, indicating that our methods are robust to the absence of prior pre-training. In Table 12, we report the RMSE and NLL results of fSGLD and fSGHMC on the UCI regression (Yacht) dataset without pre-trained GP priors. Even though the fixed GP priors without pre-training show slight degradation, the performance remains competitive compared to other baseline methods.

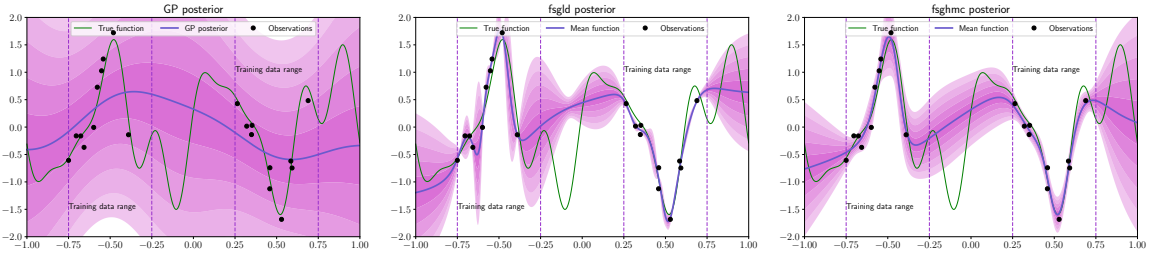
Furthermore, we analyze the influence of different pre-training epochs for the GP priors by comparing results with 100, 20, and 10 pre-training epochs. Figure 8 provides a detailed comparison: the leftmost columns, Figure 8(a), Figure 8(d) and Figure 8(g) depict the predictions of the three different GP priors, while the middle and rightmost columns illustrate the fSGLD and fSGHMC posterior results from these priors, respectively. Interestingly, while the predictions from the three GP priors differ significantly, there is no noticeable difference in the corresponding



(a) GP-100 epochs (b) fSGLD (c) fSGHMC



(d) GP-20 epochs (e) fSGLD (f) fSGHMC



(g) GP-10 epochs (h) fSGLD (i) fSGHMC

Figure 8: 1-D extrapolation example of different pre-training epochs for GP prior. The leftmost column shows the predictions of GP prior with three different pre-training epochs: 100, 20, and 10 epochs from top to bottom. The middle and the rightmost columns are the corresponding fSGLD and fSGHMC posteriors from these three GP prior, respectively.

fSGLD and fSGHMC posteriors. This observation highlights that the effectiveness of our methods does not strongly depend on the pre-training procedure of the GP prior, making them more robust and flexible in practical applications.