IMPROVING GENERALIZATION WITH FLAT HILBERT BAYESIAN INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce Flat Hilbert Bayesian Inference (FHBI), an algorithm designed to enhance generalization in Bayesian inference. Our approach involves an iterative two-step procedure with an adversarial functional perturbation step and a functional descent step within the reproducing kernel Hilbert spaces. This methodology is supported by a theoretical analysis that extends previous findings on generalization ability from finite-dimensional Euclidean spaces to infinite-dimensional functional spaces. To evaluate the effectiveness of FHBI, we conduct comprehensive comparisons against nine baseline methods on the `VTAB-1K` benchmark, which encompasses 19 diverse datasets across various domains with diverse semantics. Empirical results demonstrate that FHBI consistently outperforms the baselines by notable margins, highlighting its practical efficacy. Our code is available at `https://anonymous.4open.science/r/Flat-Hilbert-Variational-Inference-008F/`.

1 INTRODUCTION

Quantifying and tackling uncertainty in deep learning is one of the most challenging problems, mainly due to the inherent randomness of the real world and the presence of noisy data. Bayesian inference provides a robust framework for understanding complex data, allowing for probabilistic interpretation of deep learning models and reasoning under uncertainty. This approach not only facilitates predictions but also enables the quantification of uncertainty. A primary challenge in this domain is the computation and sampling from intricate distributions, mainly when dealing with deep learning models. One effective strategy to tackle this issue is variational inference, which seeks to approximate the true posterior distribution with simpler forms, known as approximate posteriors while optimizing a variational lower bound. Several techniques have been developed in this area, including those by Kingma & Welling (2013); Kingma et al. (2015), and Blundell et al. (2015), who extended the Gaussian variational posterior approximation for neural networks, as well as Gupta & Nagar (2018), who enhanced the flexibility of posterior approximations. In addition to variational methods, various particle sampling techniques have been proposed for Bayesian inference, especially in scenarios requiring multiple models. Notable particle sampling methods include Hamiltonian Monte Carlo (HMC) (Neal, 1996), Stochastic Gradient Langevin Dynamics (SGLD) (Welling & Teh, 2011), Stochastic Gradient HMC (SGHMC) (Chen et al., 2014), and Stein Variational Gradient Descent (SVGD) (Liu & Wang, 2016b). Each method contributes to a deeper understanding and more practical application of Bayesian inference in deep learning.

Besides quantifying uncertainty, tackling overfitting is a major challenge in machine learning. Overfitting often occurs when the training process gets stuck in local minima, leading to a model that fails to generalize well to unseen data. This problem is mainly due to loss functions' high-dimensional and non-convex nature, which often exhibit multiple local minima in the loss landscape. In standard deep network training, flat minimizers effectively improve model generalization (Keskar et al., 2016; Kaddour et al., 2022; Li et al., 2022). Among the flat minimizers, Sharpness-Aware Minimization (SAM) (Foret et al., 2021) has emerged as a practical approach by concurrently minimizing the empirical loss and reducing the sharpness of the loss function. Recently, SAM has demonstrated its versatility and effectiveness across a wide range of tasks, including meta-learning (Abbas et al., 2022), vision models (Chen et al., 2021), and language models (Bahri et al., 2022).

[*]These authors contributed equally to this work

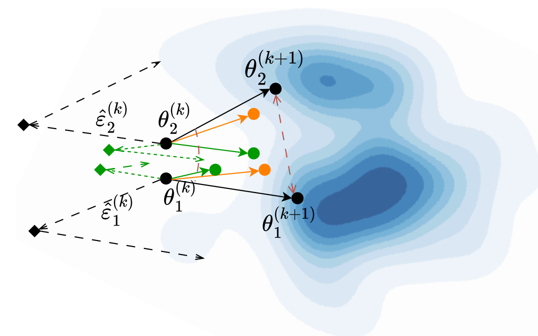


Figure 1: Schematic of SAM w. independent particles (green), SVGD (orange), and our FHBI (Algorithm 1) (black) updates. SAM's particles are not aware of other's trajectories. SVGD only seeks the modes and promotes *spatial* diversity. FHBI seeks the modes, minimizes sharpness, and promotes *spatial* and *angular* diversity.

Contribution. We bridge the gap between the flat minimizers and particle sampling to introduce a Bayesian inference framework with improved generalization ability. To accomplish this, we first present Theorem 1, which strengthens prior generalization bounds from finite-dimensional Euclidean spaces to the reproducing kernel Hilbert spaces (RKHS), which are broader and more general functional spaces that are typically infinite-dimensional. Notably, this theorem introduces the notion of *functional sharpness* that offers an insight to improve the generalization ability of current particle-sampling methods. Subsequently, Theorem 2 translates these notions of functional sharpness and generalization in RKHS into the context of Bayesian inference. This analysis establishes a connection between the general and empirical KL loss, providing a strategy to enhance generalization by minimizing the general KL loss. Motivated by these two theorems, we derive Flat Hilbert Bayesian Inference (FBVI), a practical algorithm that employs a dual-step functional sharpness-aware update procedure in RKHS. This approach improves the generalization of sampled particles, thereby enhancing the quality of the ensemble. Overall, our contributions are as follows:

1. We present a theoretical analysis that characterizes generalization ability over the functional space. This analysis generalizes prior works from the Euclidean space to infinite-dimensional functional space, thereby introducing the notion of *functional sharpness* i.e., the sharpness of the functional spaces.
2. Building on this theoretical foundation, we propose a practical particle-sampling algorithm that enhances the generalization ability over existing methods. We conducted extensive experiments comparing our Flat Hilbert Bayesian Inference (FHBI) algorithm with nine baselines on the `VTAB-1K` benchmark, which includes 19 datasets across various domains and semantics. Experimental results demonstrated that our algorithm outperforms these baselines by notable margins.

The paper is structured as follows: Section 2 reviews the related works on Bayesian inference and the development of flat minimizers. Section 3 provides the necessary background and notations. Section 4 discusses the motivation and theoretical development behind our sharpness-aware particle-sampling approach. Section 5 presents experimental results, comparing our algorithm against various Bayesian inference baselines across diverse settings. Section 6 offers a deeper analysis of FHBI's behavior to gain further insight into its effectiveness over the baseline methods.

2 Related Works

Sharpness-aware minimization. Flat minimizers have been shown to be more robust to the shifts between training and test losses, thereby enhancing the generalization ability of neural networks (Jiang et al., 2020; Petzka et al., 2021; Dziugaite & Roy, 2017). The relationship between generalization and the width of minima has been studied both theoretically and empirically in several prior works (Hochreiter & Schmidhuber, 1994; Neyshabur et al., 2017; Dinh et al., 2017; Fort & Ganguli, 2019). Consequently, a variety of methods have been developed to search for flat minima (Pereyra et al., 2017; Chaudhari et al., 2017; Keskar et al., 2017; Izmailov et al., 2018).

Among the flat minizer, Sharpness-Aware Minimization (SAM), introduced by Foret et al. (2021), has gained significant attention due to its effectiveness and scalability. SAM's versatility has been leveraged across a wide range of tasks and domains, including domain generalization (Cha et al., 2021; Wang et al., 2023; Zhang et al., 2023), federated learning (Caldarola et al., 2022; Qu et al., 2022), Bayesian networks (Nguyen et al., 2023a; Möllenhoff & Khan, 2023), and meta-learning (Abbas et al., 2022). Moreover, SAM has demonstrated its ability to enhance generalization in both vision models (Chen et al., 2021) and language models (Bahri et al., 2022).

Nevertheless, these studies are constrained to finite-dimensional Euclidean spaces. In this work, we strengthen these generalization principles to infinite-dimensional functional spaces and propose a particle-sampling method grounded in this theoretical framework.

Bayesian Inference. Two main strategies were widely employed in the literature of Bayesian inference. The first paradigm is *Variational Inference*, which aims to approximate a target distribution by selecting a distribution from a family of potential approximations and optimizing a variational lower bound. Graves (2011) introduced the use of a Gaussian variational posterior approximation for neural network weights, which was later extended in Kingma & Welling (2013); Kingma et al. (2015); Blundell et al. (2015) with the reparameterization trick to facilitate training deep latent variable models. Louizos & Welling (2017) proposed using a matrix-variate Gaussian to model entire weight matrices (Gupta & Nagar, 2018) to increase further the flexibility of posterior approximations, which offers a novel approach to approximate the posterior. Subsequently, various alternative structured forms of the variational Gaussian posterior were proposed, including the Kronecker-factored approximations (Zhang et al., 2018; Ritter et al., 2018; Rossi et al., 2020), or non-centered or rank-1 parameterizations (Ghosh et al., 2018; Dusenberry et al., 2020).

The second paradigm in the literature of Bayesian inference is *Markov Chain Monte Carlo* (MCMC), which involves sampling multiple models from the posterior distribution. MCMC has been applied to neural network inference, such as Hamiltonian Monte Carlo (HMC) (Neal, 1996). However, HMC requires the computation of full gradients, which can be computationally expensive. To address this, Stochastic Gradient Langevin Dynamics (SGLD) (Welling & Teh, 2011) integrates first-order Langevin dynamics within a stochastic gradient framework. Stochastic Gradient HMC (SGHMC) (Chen et al., 2014) further incorporates stochastic gradients into Bayesian inference, enabling scalability and efficient exploration of different solutions. Another critical approach, Stein Variational Gradient Descent (SVGD) (Liu & Wang, 2016a), closely related to our work, uses a set of particles that converge to the target distribution. It is also theoretically established that SGHMC, SGLD, and SVGD asymptotically sample from the posterior as the step sizes approach zero.

3 Backgrounds and Notations

Bayesian Inference. Consider a family of neural networks $f_{\boldsymbol{\theta}}(x)$, where the random variable $\boldsymbol{\theta}$ represents the model parameters and takes values in the model space $\Theta \subset \mathbb{R}^d$. We are given a training set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^n$ of n i.i.d observations from the data space $\mathcal{X} \times \mathcal{Y}$, and the prior distribution of the parameters $p(\boldsymbol{\theta})$. In the literature on Bayesian inference problems, prior works typically focus on approximating the *empirical posterior* $\mathbb{P}_{\theta|\mathcal{S}}$, whose density function $p(\boldsymbol{\theta}|\mathcal{S})$ is defined as:

$$p(\boldsymbol{\theta}|\mathcal{S}) \propto p(\boldsymbol{\theta}) \prod_{i=1}^{n} p(y_i|x_i, \mathcal{S}, \boldsymbol{\theta}),$$

where the prior distribution $\mathbb{P}_\theta$ has the density function $p(\boldsymbol{\theta})$. The likelihood term is proportional to

$$p(y|x, \mathcal{S}, \boldsymbol{\theta}) \propto \exp\left(-\frac{1}{|\mathcal{S}|}\ell(f_{\boldsymbol{\theta}}(x), y)\right) = \exp\left(-\frac{1}{n}\ell(f_{\boldsymbol{\theta}}(x), y)\right),$$

with some loss function ℓ and a sufficiently expressive model f_θ. Then, the empirical posterior is:

$$p(\boldsymbol{\theta}|\mathcal{S}) \propto \exp\left(-\frac{1}{n}\sum_{i=1}^{n}\ell(f_{\boldsymbol{\theta}}(x_i), y_i)\right) p(\boldsymbol{\theta}). \tag{1}$$

More formally, the empirical posterior is equal to:

$$p(\boldsymbol{\theta}|\mathcal{S}) = \exp\left(-\frac{1}{n}\sum_{i=1}^{n}\ell(f_{\boldsymbol{\theta}}(x_i), y_i)\right)p(\boldsymbol{\theta})/Z_{\mathcal{S}}, \tag{2}$$

where $Z_{\mathcal{S}}$ is the normalizing constant. We define the population and empirical losses as follows:

$$\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(f_{\boldsymbol{\theta}}(x), y)],$$
$$\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}) = \mathbb{E}_{(x,y)\sim\mathcal{S}}[\ell(f_{\boldsymbol{\theta}}(x), y)] = \frac{1}{n}\sum_{i=1}^{n}\ell(f_{\boldsymbol{\theta}}(x_i), y_i).$$

The *population loss* is defined as the expected loss over the entire data-label distribution. In contrast, the *empirical loss* is the average loss computed over a given training set $\mathcal{S}$. Based on these definitions, the empirical posterior in Eq. 2 can be written as:

$$p(\boldsymbol{\theta}|\mathcal{S}) = \exp(-\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}))p(\boldsymbol{\theta})/Z_{\mathcal{S}}.$$

Intuitively, models with parameters $\boldsymbol{\theta}$ that fit well to the training set $\mathcal{S}$ lead to lower empirical loss values, resulting in higher density in the empirical posterior. However, simply fitting to the training samples can lead to overfitting. To improve generalization, we are more concerned with performance over the entire data distribution $\mathcal{D}$ rather than just the specific sample $\mathcal{S}$. Accordingly, we define the population posterior as $\mathbb{P}_{\mathcal{D}}$ whose density is given by:

$$p(\boldsymbol{\theta}|\mathcal{D}) = \exp(-\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}))p(\boldsymbol{\theta})/Z_{\mathcal{D}}, \tag{3}$$

with the normalizing constant $Z_{\mathcal{D}}$. This population posterior is more general than the empirical posterior, as it captures the true posterior of the parameters under the full data distribution. However, understanding the population posterior is particularly challenging because we can only access the empirical loss $\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$, not the population loss $\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta})$. In this paper, we deviate from prior approaches that primarily focus on approximating the empirical posterior and instead propose a particle-sampling method to approximate the population posterior.

Reproducing Kernel Hilbert Space (RKHS). Let $k(\boldsymbol{\theta}, \boldsymbol{\theta}') : \Theta \times \Theta \to \mathbb{R}$ be a positive definite kernel operating on the model space. The reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ of $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ is the closure of the linear span $\{f : f(\cdot) = \sum_i a_i k(\cdot, \boldsymbol{\theta}_i), a_i \in \mathbb{R}, \boldsymbol{\theta}_i \in \Theta\}$. For $f(\boldsymbol{\theta}) = \sum_i a_i k(\boldsymbol{\theta}, \boldsymbol{\theta}_i)$ and $g(\boldsymbol{\theta}) = \sum_j b_j k(\boldsymbol{\theta}, \boldsymbol{\theta}_j)$, $\mathcal{H}$ is equipped with the inner product defined by $\langle f, g\rangle_{\mathcal{H}} = \sum_{ij} a_i b_j k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$. For all $\boldsymbol{\theta} \in \Theta$, there exists a unique element $K_{\boldsymbol{\theta}} \in \mathcal{H}$ with the reproducing property that $f(\boldsymbol{\theta}) = \langle f, K_{\boldsymbol{\theta}}\rangle_{\mathcal{H}}$ for any $f \in \mathcal{H}$.

Given that $\mathcal{H}$ is a scalar-valued RKHS with kernel $k(\theta, \theta')$, $\mathcal{H}^d = \mathcal{H} \times \mathcal{H} \times \cdots \times \mathcal{H}$ is a vector-valued RKHS of functions $\boldsymbol{f} = [f_1, f_2, \cdots, f_d]$ corresponding to the kernel $K(\boldsymbol{\theta}, \boldsymbol{\theta}') = k(\boldsymbol{\theta}, \boldsymbol{\theta}')\boldsymbol{I}$. $\mathcal{H}^d$ is equipped with the inner product $\langle \boldsymbol{f}, \boldsymbol{g}\rangle_{\mathcal{H}^d} = \sum_{i=1}^{d}\langle f_i, g_i\rangle_{\mathcal{H}}$.

Let $\boldsymbol{F}[\boldsymbol{f}]$ be a functional on $\boldsymbol{f} \in \mathcal{H}^d$. Similar to the definition by Liu & Wang (2016b), the (functional) gradient of $\boldsymbol{F}$ is defined as a function $\nabla_{\boldsymbol{f}}\boldsymbol{F}[\boldsymbol{f}] \in \mathcal{H}^d$ such that for any $\boldsymbol{g} \in \mathcal{H}^d$ and $\epsilon \in \mathbb{R}$

$$\boldsymbol{F}[\boldsymbol{f} + \epsilon \boldsymbol{g}] = \boldsymbol{F}[\boldsymbol{f}] + \epsilon\,\langle \nabla_{\boldsymbol{f}}\boldsymbol{F}[\boldsymbol{f}], g\rangle_{\mathcal{H}^d} + \mathcal{O}(\epsilon^2). \tag{4}$$

Stein Variational Gradient Descent (SVGD). Given a general target distribution $p(\boldsymbol{\theta})$, SVGD (Liu & Wang, 2016b) aims to find a flow of distributions $\{q^{(k)}\}_k$ that minimizes the KL distance to the target distribution. Motivated by the Stein identity and Kernelized Stein Discrepancy, SVGD proposes the update $q^{(k+1)} = q^{(k)}_{[\boldsymbol{T}]}$, in which $\boldsymbol{T} : \Theta \to \Theta$ is a smooth one-to-one push-forward map of the form $\boldsymbol{T}(\boldsymbol{\theta}) = \boldsymbol{\theta} + \epsilon\phi^*_{p,q}(\boldsymbol{\theta})$ in which:

$$\phi^*_{p,q}(\cdot) = \mathbb{E}_{\boldsymbol{\theta}\sim q}[\mathcal{A}_p k(\boldsymbol{\theta}, \cdot)] \qquad \text{and} \qquad \mathcal{A}_p\phi(\boldsymbol{\theta}) = \phi(\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}\log p(\boldsymbol{\theta})^\top + \nabla_{\boldsymbol{\theta}}\phi(\boldsymbol{\theta}).$$

Here, $\mathcal{A}_p$ is known as the Stein operator, which acts on ϕ and produces a zero-mean function $\mathcal{A}_p\phi(\boldsymbol{\theta})$ when $\boldsymbol{\theta} \sim p$. Notably, while SVGD is designed for general target distributions p, in the context of Bayesian inference, it is only applicable to the empirical posterior rather than the population posterior, which we will discuss in detail in the next section.

4 Flat Hilbert Bayesian Inference (FHBI)

Consider the Bayesian inference problem of approximating a posterior distribution. In prior works, such as SVGD (Liu & Wang, 2016b), when applying to the context of Bayesian inference, the methods are only applicable to the *empirical posterior* $p(\boldsymbol{\theta}|\mathcal{S})$ because we only have access to the empirical loss. It is evident that when sampling a set of m particle models $\boldsymbol{\theta}_{1:m}$ from $p(\boldsymbol{\theta}|\mathcal{S})$, these particles congregate in the high-density regions of the empirical posterior $p(\boldsymbol{\theta}|\mathcal{S})$, corresponding to the areas with low *empirical loss* $\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$. However, to avoid overfitting, it is preferable to sample the particle models $\boldsymbol{\theta}_{1:m}$ from the *population posterior* $p(\boldsymbol{\theta}|\mathcal{D}) \propto \exp(-\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}))p(\boldsymbol{\theta})$, as this approach directs the particle models $\boldsymbol{\theta}_{1:m}$ towards regions with low values of the *population loss* $\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta})$, thus improving generalization ability. To better understand this motivation from a theoretical perspective, consider the following proposition, with the proof provided in Appendix A.1:

Proposition 1. *Consider the problem of finding the distribution* $\mathbb{Q}$ *that solves:*

$$\mathbb{Q}^* = \min_{\mathbb{Q} \ll \mathbb{P}_\theta} \left\{ \mathbb{E}_{\theta \sim \mathbb{Q}}[\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta})] + D_{\mathrm{KL}}(\mathbb{Q} \| \mathbb{P}_\theta) \right\} \tag{5}$$

where we search over $\mathbb{Q}$ *absolutely continuous w.r.t* $\mathbb{P}_{\boldsymbol{\theta}}$*, and the second term is the regularization term. The closed-form solution to this problem is exactly the population posterior defined in Eq. 3.*

In this proposition, our aim is to identify the posterior distribution that minimizes the *expected population loss*, where the expectation is taken over the entire parameter space with $\boldsymbol{\theta} \sim \mathbb{Q}^*$, while maintaining proximity to the prior to ensure simplicity. With access to this posterior $\mathbb{Q}^*$, we can sample a set of particles whose average performance optimally minimizes the population loss. Since the solution to this optimization problem corresponds exactly to the population posterior, the *ensemble of the particles* sampled from $\mathbb{Q}^* \equiv p(\theta|\mathcal{D})$ effectively minimizes the average value of the population loss. This is because $\mathbb{Q}^*$ is explicitly chosen to minimize the expected value of the population loss $\mathcal{L}_{\mathcal{D}}$, which means the ensemble fits the whole data distribution instead of overfitting to the specific dataset $\mathcal{S}$, therefore establishes improved generalizability. Consequently, this proposition theoretically asserts that sampling from $p(\boldsymbol{\theta}|\mathcal{D})$ improves the generalizability of the ensemble.

4.1 Theoretical Analysis

Motivated by this observation, we seek to advance prior work by *approximating the general posterior*. Specifically, to improve generalizability, our objective is to approximate the target general posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ using a simpler distribution $q^*(\boldsymbol{\theta})$ drawn from a predefined set of distributions $\mathcal{F}$. This is achieved by minimizing the KL divergence:

$$q^* = \arg\min_{q \in \mathcal{F}} D_{\mathrm{KL}} \Bigg(q(\boldsymbol{\theta}) \| p(\boldsymbol{\theta}|\mathcal{D}) \Bigg). \tag{6}$$

Ideally, the set $\mathcal{F}$ should be simple enough for a simple solution and effective inference while sufficiently broad to approximate a wide range of target distributions closely. Let $q(\boldsymbol{\theta})$ be the density of a reference distribution. We define $\mathcal{F}$ as the set of distributions for random variables of the form $\vartheta = \boldsymbol{T}(\boldsymbol{\theta})$, where $\boldsymbol{T} : \Theta \to \Theta$ is a smooth, bijective mapping, and $\boldsymbol{\theta}$ is sampled from q. By variable change, the density of ϑ, denoted as $q_{[\boldsymbol{T}]}(\cdot)$, is expressed as follows:

$$q_{[\boldsymbol{T}]}(\vartheta) = q(\boldsymbol{T}^{-1}(\boldsymbol{\theta}))|\det(\nabla_\vartheta \boldsymbol{T}^{-1}(\vartheta))|.$$

We restrict the set of the smooth transformations $\boldsymbol{T}$ to the set of push-forward maps of the form $\boldsymbol{T}(\boldsymbol{\theta}) = \boldsymbol{\theta} + \boldsymbol{f}(\boldsymbol{\theta})$, where $\boldsymbol{f} \in \mathcal{H}^d$. When $\|\boldsymbol{f}\|_{\mathcal{H}^d}$ is sufficiently small, the Jacobian of $\boldsymbol{T} = \boldsymbol{I} + \boldsymbol{f}$ is full-rank where $\boldsymbol{I}$ denotes the identity map, in which case $\boldsymbol{T}$ is guaranteed to be a one-to-one map according to the inverse function theorem. Under this restriction, the problem is equivalent to solving an optimization problem over the RKHS:

$$\boldsymbol{f}^* = \arg\min_{\boldsymbol{f} \in \mathcal{H}^d, \|\boldsymbol{f}\|_{\mathcal{H}^d} \le \epsilon} D_{\mathrm{KL}} \Bigg(q_{[\boldsymbol{I}+\boldsymbol{f}]}(\boldsymbol{\theta}) \| p(\boldsymbol{\theta}|\mathcal{D}) \Bigg).$$

The challenge with this optimization problem lies in our lack of access to the general loss function $\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta})$ and the general posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$. We present our first theorem to address this issue, which characterizes generalization ability in the functional space $\mathcal{H}^d$. The proof of this theorem can be found in Appendix A.2.

Theorem 1 (Informal)**.** *Let $\tilde{\ell} : \mathcal{H}^d \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$ be a loss function on the RKHS $\mathcal{H}^d$ and the data space. Define $\tilde{L}_{\mathcal{D}}(\boldsymbol{f}) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\tilde{\ell}(\boldsymbol{f}, x, y)]$ and $\tilde{L}_{\mathcal{S}}(\boldsymbol{f}) = \frac{1}{n}\sum_{i=1}^{n} \tilde{\ell}(\boldsymbol{f}, x_i, y_i)$ be the corresponding general and empirical losses. Then for any $\rho > 0$ and any distribution $\mathcal{D}$, with probability of $1-\delta$ over the choice of the training set $\mathcal{S} \sim \mathcal{D}^n$, we have:*

$$\tilde{L}_{\mathcal{D}}(\boldsymbol{f}) \leq \max_{\boldsymbol{f}' \in \mathcal{H}^d, \|\boldsymbol{f}'-\boldsymbol{f}\|_{\mathcal{H}^d} \leq \rho} \tilde{L}_{\mathcal{S}}(\boldsymbol{f}') + \mathcal{O}\left(\sqrt{\frac{\log(1+\frac{1}{\rho^2}) + \log\left(\frac{n}{\delta}\right)}{n-1}}\right),$$

This theorem extends prior results, such as the generalization bounds established by Foret et al. (2021) and Kim et al. (2022), from Euclidean space to a broader, more general reproducing kernel Hilbert space. It is noteworthy that this is not a straightforward extension, as the previous generalization bounds rely on the dimensionality of the domain, while the RKHS is typically infinite-dimensional for many widely used kernels such as the RBF kernels (Aronszajn, 1950). Building on the first theorem, we present the second theorem, which directly addresses the general posterior and serves as the primary motivation for our method. The proof of this theorem can be found in Appendix A.3.

Theorem 2 (Informal)**.** *Assume that q is any distribution. For any $\rho > 0$, with probability of $1-\delta$ over the training set $\mathcal{S}$ generated by distribution $\mathcal{D}$, we have:*

$$D_{\mathrm{KL}}\left(q_{[\boldsymbol{I}+\boldsymbol{f}]} || p(\boldsymbol{\theta}|\mathcal{D})\right) \leq \max_{\boldsymbol{f}' \in \mathcal{H}^d, \|\boldsymbol{f}'-\boldsymbol{f}\| \leq \rho} D_{\mathrm{KL}}\left(q_{[\boldsymbol{I}+\boldsymbol{f}']} || p(\boldsymbol{\theta}|\mathcal{S})\right) + \mathcal{O}\left(\sqrt{\frac{\log(1+\frac{1}{\rho^2}) + \log\left(\frac{n}{\delta}\right)}{n-1}}\right).$$

Our objective is to learn the function $\boldsymbol{f}^* \in \mathcal{H}^d$ that minimizes $D_{\mathrm{KL}}(q_{[\boldsymbol{I}+\boldsymbol{f}]} \| p(\boldsymbol{\theta}|\mathcal{D}))$. Motivated by Theorem 2, we propose to *implicitly* minimize $D_{\mathrm{KL}}(q_{[\boldsymbol{I}+\boldsymbol{f}]} || p(\boldsymbol{\theta}|\mathcal{D}))$ by minimizing the right-hand side term $\max_{\|\boldsymbol{f}'-\boldsymbol{f}\|_{\mathcal{H}^d} \leq \rho} D_{\mathrm{KL}}\left(q_{[\boldsymbol{I}+\boldsymbol{f}']} || p(\boldsymbol{\theta}|\mathcal{S})\right)$. For any $\boldsymbol{f} \in \mathcal{H}^d$, let $\boldsymbol{F}[\boldsymbol{f}] = D_{\mathrm{KL}}\left(q_{[\boldsymbol{I}+\boldsymbol{f}]} \| p(\boldsymbol{\theta}|\mathcal{S})\right)$ and $\boldsymbol{f}' = \boldsymbol{f} + \rho\hat{\boldsymbol{f}}$, it follows that:

$$\arg\max_{\|\boldsymbol{f}'-\boldsymbol{f}\|_{\mathcal{H}^d} \leq \rho} D_{\mathrm{KL}}\left(q_{[\boldsymbol{I}+\boldsymbol{f}']} || p(\boldsymbol{\theta}|\mathcal{S})\right) = \arg\max_{\|\hat{\boldsymbol{f}}\|_{\mathcal{H}^d} \leq 1} \boldsymbol{F}[\boldsymbol{f} + \rho\hat{\boldsymbol{f}}] \tag{7}$$

$$= \arg\max_{\|\hat{\boldsymbol{f}}\|_{\mathcal{H}^d} \leq 1} \boldsymbol{F}[\boldsymbol{f}] + \rho\left\langle \hat{\boldsymbol{f}}, \nabla_{\boldsymbol{f}} \boldsymbol{F}[\boldsymbol{f}] \right\rangle_{\mathcal{H}^d} + \mathcal{O}(\rho^2) \approx \arg\max_{\|\hat{\boldsymbol{f}}\|_{\mathcal{H}^d} \leq 1} \left\langle \hat{\boldsymbol{f}}, \nabla_{\boldsymbol{f}} \boldsymbol{F}[\boldsymbol{f}] \right\rangle_{\mathcal{H}^d}. \tag{8}$$

Let $\boldsymbol{g} = \nabla_{\boldsymbol{f}} \boldsymbol{F}[\boldsymbol{f}] \in \mathcal{H}^d$. The Cauchy-Schwarz inequality on Hilbert spaces (Kreyszig, 1978) implies:

$$\left| \left\langle \hat{\boldsymbol{f}}, \boldsymbol{g} \right\rangle_{\mathcal{H}^d} \right| \leq \left\langle \hat{\boldsymbol{f}}, \boldsymbol{g} \right\rangle_{\mathcal{H}^d} \leq \|\hat{\boldsymbol{f}}\|_{\mathcal{H}^d} \|\boldsymbol{g}\|_{\mathcal{H}^d} \leq \|\boldsymbol{g}\|_{\mathcal{H}^d}.$$

In turn, the solution $\hat{\boldsymbol{f}}^*$ that solves the maximization problem in Eq. 8 is given by:

$$\hat{\boldsymbol{f}}^* = \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|_{\mathcal{H}^d}} = \frac{\nabla_{\boldsymbol{f}} D_{\mathrm{KL}}\left(q_{[\boldsymbol{I}+\boldsymbol{f}]} \| p(\cdot|\mathcal{S})\right)}{\left\| \nabla_{\boldsymbol{f}} D_{\mathrm{KL}}\left(q_{[\boldsymbol{I}+\boldsymbol{f}]} \| p(\cdot|\mathcal{S})\right) \right\|_{\mathcal{H}^d}}. \tag{9}$$

Recall that our goal is to find a sequence of functions $\{\boldsymbol{f}_k\}_k \subset \mathcal{H}^d$ that converges toward the optimal solution $\boldsymbol{f}^*$. With the sequence $\{\boldsymbol{f}_k\}_k$, we can obtain the flow of distributions $\{q^{(k)}\}_k$, in which $q^{(k)} = q_{[\boldsymbol{I}+\boldsymbol{f}_k]}$, that gradually approaches the optimal solution of Eq. 6. Motivated by Eq. 9, we propose the following *functional sharpness-aware* update procedure:

$$\hat{\boldsymbol{f}}_k^* = \rho \frac{\nabla_{\boldsymbol{f}} D_{\mathrm{KL}}\left(q_{[\boldsymbol{I}+\boldsymbol{f}]} \| p(\cdot|\mathcal{S})\right)\Big|_{\boldsymbol{f}=\boldsymbol{f}_k}}{\left\| \nabla_{\boldsymbol{f}} D_{\mathrm{KL}}\left(q_{[\boldsymbol{I}+\boldsymbol{f}]} \| p(\cdot|\mathcal{S})\right)\Big|_{\boldsymbol{f}=\boldsymbol{f}_k} \right\|_{\mathcal{H}^d}}, \qquad \text{(Functional Ascend step)} \tag{10}$$

$$\boldsymbol{f}_{k+1} = \boldsymbol{f}_k - \epsilon \nabla_{\boldsymbol{f}} D_{\mathrm{KL}}\left(q_{[\boldsymbol{I}+\boldsymbol{f}]} \| p(\cdot|\mathcal{S})\right)\Big|_{\boldsymbol{f}=\boldsymbol{f}_k+\hat{\boldsymbol{f}}_k^*}, \qquad \text{(Functional Descending step)} \tag{11}$$

$$q^{(k+1)} = q_{[\boldsymbol{I}+\boldsymbol{f}_{k+1}]}. \qquad \text{(Distributional Transformation)} \tag{12}$$

Algorithm 1 FLAT HILBERT BAYESIAN INFERENCE (FHBI)

Input: Initial particles $\{\boldsymbol{\theta}_i^{(0)}\}_{i=1}^m$, number of epochs N, step size $\rho > 0$
Output: A set of particles $\{\boldsymbol{\theta}_i\}_{i=1}^m$ that approximates the general posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$
for iteration k **do**
  $\hat{\varepsilon}_i^{(k)} \leftarrow \rho \frac{\phi(\boldsymbol{\theta}_i^{(k)})}{\|\phi(\boldsymbol{\theta}_i^{(k)})\|}$ where $\phi(\boldsymbol{\theta}) = -\frac{1}{n}\sum_{j=1}^m [k(\boldsymbol{\theta}, \boldsymbol{\theta}_j^{(k)}) \nabla_{\boldsymbol{\theta}_j^{(k)}} \log p(\boldsymbol{\theta}_j^{(k)}|\mathcal{S}) + \nabla_{\boldsymbol{\theta}_j^{(k)}} k(\boldsymbol{\theta}, \boldsymbol{\theta}_j^{(k)})]$
  $\boldsymbol{\theta}_i^{(k+1)} \leftarrow \boldsymbol{\theta}_i^{(k)} - \epsilon_i \psi(\boldsymbol{\theta}_i^{(k)}, \hat{\varepsilon}_i^{(k)})$
where $\psi(\boldsymbol{\theta}, \varepsilon) = -\frac{1}{n}\sum_{j=1}^m [k(\boldsymbol{\theta}, \boldsymbol{\theta}_j^{(k)}) \nabla_{\boldsymbol{\theta}_j^{(k)}} \log p(\boldsymbol{\theta}_j^{(k)} + \varepsilon|\mathcal{S}) + \nabla_{\boldsymbol{\theta}_j^{(k)}} k(\boldsymbol{\theta}, \boldsymbol{\theta}_j^{(k)})]$.
end for

To implement this iterative procedure, we must work with the functional gradient terms. For this, we rely on the following lemma, with the proof provided in Appendix B of (Liu & Wang, 2016b):

Lemma 1. *Let $\boldsymbol{F}[\boldsymbol{f}] = D_{\mathrm{KL}}(q_{[\boldsymbol{I}+\boldsymbol{f}]} \| p(\cdot|\mathcal{S}))$. When $\|\boldsymbol{f}\|$ is sufficiently small,*

$$\nabla_{\boldsymbol{f}} \boldsymbol{F}[\boldsymbol{f}] = -\mathbb{E}_q[\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta} + \boldsymbol{f}(\boldsymbol{\theta})|\mathcal{S}) k(\boldsymbol{\theta}, \cdot) + (I + \nabla_{\boldsymbol{\theta}} \boldsymbol{f}(\boldsymbol{\theta}))^{-1} \nabla_{\boldsymbol{\theta}} k(\boldsymbol{\theta}, \cdot)] \tag{13}$$

$$\approx -\mathbb{E}_q[\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta} + \boldsymbol{f}(\boldsymbol{\theta})|\mathcal{S}) k(\boldsymbol{\theta}, \cdot) + \nabla_{\boldsymbol{\theta}} k(\boldsymbol{\theta}, \cdot)] \stackrel{\text{def}}{=} \boldsymbol{D}(\boldsymbol{f}). \tag{14}$$

Substituting equation (14) into equations (10) and (11), the iterative procedure described from equations (10)-(11) becomes:

$$\hat{\boldsymbol{f}}_k^* = \rho \frac{\boldsymbol{D}(\boldsymbol{f}_k)}{\|\boldsymbol{D}(\boldsymbol{f}_k)\|_{\mathcal{H}^d}}, \tag{15}$$

$$\boldsymbol{f}_{k+1} = \boldsymbol{f}_k - \epsilon \boldsymbol{D}(\boldsymbol{f}_k + \hat{\boldsymbol{f}}_k^*), \tag{16}$$

$$q^{(k+1)} = q_{[\boldsymbol{I}+\boldsymbol{f}_{k+1}]}. \tag{17}$$

Even though we do not have access to $p(\boldsymbol{\theta}|\mathcal{S})$, we can compute $\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}|\mathcal{S})$ because $\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}|\mathcal{S}) = \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$. To implement the procedure above, we first draw a set of m particles $\{\boldsymbol{\theta}_i^{(0)}\}_{i=1}^m$ on the model space from the initial density, and then iteratively update the particles with an empirical version of $\boldsymbol{D}(\boldsymbol{f})$. Consequently, we obtain the practical procedure summarized in Algorithm 1, which deterministically transports the set of particles to match the empirical posterior distribution $p(\boldsymbol{\theta}|\mathcal{S})$, therefore match the general posterior $p(\boldsymbol{\theta}|\mathcal{D})$ as supported by Theorem 2. In Algorithm 1, at each iteration k, we have m particles $\{\boldsymbol{\theta}_j^{(k)}\}_{j=1}^m$. Eq. 15 computes the m ascend steps $\hat{\varepsilon}_i^{(k)}$; then, Eq. 16 and Eq. 17 use these ascend steps to transport the m model particles to $\{\boldsymbol{\theta}_j^{(k+1)}\}_{j=1}^m$. It is noteworthy that FHBI is a generalization of both SVGD and SAM. In particular, if we set $\rho = 0$, we get SVGD; when $m = \#\text{PARTICLES} = 1$, we obtain SAM.

Interactive gradient directions and Connections to SAM. To gain further insight into the mechanism of FHBI and its underlying connections to SAM, consider the term $\nabla_{\boldsymbol{\theta}_j} \log(\boldsymbol{\theta}_j + \hat{\varepsilon}_i)$ in the descending step, which is related to $\nabla_{\boldsymbol{\theta}_j} \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}_j + \hat{\varepsilon}_i)$.

The perturbed loss can be approximated as:

$$\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}_j + \hat{\varepsilon}_i) \approx \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}_j) + \hat{\varepsilon}_i \nabla_{\boldsymbol{\theta}_j} \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}_j),$$

where $\hat{\varepsilon}_i$ involves the average $\sum_{k=1}^m k(\boldsymbol{\theta}_k, \boldsymbol{\theta}_j) \nabla_{\boldsymbol{\theta}_k} \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}_k)$. Consequently, the gradient of this perturbed loss indicates a direction that simultaneously minimizes $\|\nabla_{\boldsymbol{\theta}_j} \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}_j)\|^2$ - which approximates the sharpness of the $j-$th particle, as discussed by Foret et al. (2021) - and $\nabla_{\boldsymbol{\theta}_j} \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}_j) \cdot \nabla_{\boldsymbol{\theta}_k} \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}_k)$ for all j, k, which reflects the angular similarity in the directions of the two particles. Thus, in addition to minimizing the sharpness of each particle, the first term of the descent step acts as an *angular repulsive force*, promoting more diverse traveling directions for the particles. Besides, as discussed by Liu & Wang (2016b), the second term acts as a *spatial repulsive force*, driving the particles apart to prevent them from collapsing into a single mode. Consequently, FHBI is not merely an extension of SAM to multiple independent particles; it enables the sharpness and gradient directions of the particles to interact with one another. This insight about the mechanism underlying our algorithm is summarized in Figure 1. In Section 6, we empirically demonstrate that,

	Natural							Specialized				Structured								
Method	CIFAR100	Caltech101	DTD	Flower102	Pets	SVHN	Sun397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	KITTI	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Ele	**AVG**
FFT	68.9	87.7	64.3	97.2	86.9	**87.4**	38.8	79.7	95.7	84.2	73.9	56.3	58.6	41.7	65.5	57.5	46.7	25.7	29.1	65.6
AdamW	67.1	90.7	68.9	98.1	90.1	84.5	54.2	84.1	94.9	84.4	73.6	82.9	69.2	49.8	78.5	**75.7**	47.1	31.0	**44.0**	72.0
SAM	72.7	90.3	71.4	99.0	90.2	84.4	52.4	82.0	92.6	84.1	74.0	76.7	68.3	47.9	74.3	71.6	43.4	26.9	39.1	70.5
DeepEns	69.1	88.9	67.7	98.9	90.7	85.1	54.5	82.6	94.8	82.7	75.3	46.6	47.1	47.4	68.2	71.1	36.6	30.1	35.6	67.0
BayesTune	67.2	91.7	69.5	99.0	90.7	86.4	54.7	84.9	**95.3**	84.1	75.1	**82.8**	68.9	49.7	79.3	74.3	46.6	30.3	42.8	72.2
SGLD	68.7	91.0	67.0	98.6	89.3	83.0	51.6	81.2	93.7	83.2	76.4	80.0	70.1	48.2	76.2	71.1	39.3	31.2	38.4	70.4
SADA-JEM	70.3	91.9	70.2	98.2	91.2	85.6	54.7	84.3	94.1	83.4	77.0	79.9	72.1	51.6	79.4	70.7	45.3	29.6	40.1	72.1
SA-BNN	65.1	91.5	71.0	98.9	89.4	89.3	55.2	83.2	94.5	86.4	75.2	61.4	63.2	40.0	71.3	64.5	34.5	27.2	31.2	68.1
SVGD	71.3	90.2	71.0	98.7	90.2	84.3	52.7	83.4	93.2	86.7	75.1	75.8	70.7	49.6	79.9	69.1	41.2	30.6	33.1	70.9
FHBI	**74.1**	**93.0**	**74.3**	**99.1**	**92.4**	87.3	**56.5**	**85.3**	95.0	**87.2**	**79.6**	80.1	**72.3**	**52.2**	**80.4**	72.8	**51.2**	**31.9**	41.3	**73.7**
	(.17)	(.42)	(.15)	(0.20)	(0.21)	(.52)	(.12)	(.31)	(.57)	(.21)	(.20)	(.16)	(.27)	(.47)	(.31)	(.50)	(.32)	(.36)	(.59)	

Table 1: `VTAB-1K` classification accuracy results. All the methods are applied to finetune the same set of LoRA parameters on ViT-B/16 pre-trained with `ImageNet-21K` dataset.

compared to SVGD, FHBI not only effectively minimizes particle-wise sharpness and loss values but also fosters greater diversity in the travel directions of the particles during training. This increased directional diversity, combined with the kernel gradient term, further mitigates the risk of particles collapsing into a single mode and improve the final performance as presented in Section 5.

5 Experiments

Applications to Model Fine-tuning. Bayesian inference methods have promising applications in model finetuning. In standard finetuning scenarios, we are given a pre-trained model Φ. The objective is to find the optimal parameters $\boldsymbol{\theta} = \Phi + \beta$, where β represents an additional module, often lightweight and small relative to the full model. Several parameter-efficient finetuning strategies have been developed, including LoRA (Hu et al., 2021), Adapter (Houlsby et al., 2019), and others. Our experiments focus on finetuning the ViT-B/16 architecture (Dosovitskiy et al., 2021), pre-trained with the `ImageNet-21K` dataset (Deng et al., 2009), where β is defined by the LoRA framework. For the Bayesian approaches, we aim to learn m LoRA particles $\beta^{(i)}$ to obtain m model instances $\boldsymbol{\theta}^{(i)}$. The final output is then computed as the average of the outputs from all these model instances.

Experimental Details. To assess the effectiveness of FHBI, we conduct experiments on the `VTAB-1K` benchmark (Zhai et al., 2020), a challenging image classification/prediction suite consisting of 19 datasets from various domains. `VTAB-1K` covers various tasks across different semantics and object categories. The datasets are organized into Natural, Specialized, and Structured domains. Each dataset includes 1,000 training examples, with an official 80/20 train-validation split. We compared FHBI against nine baselines with three deterministic finetuning strategies including full finetuning, AdamW, and SAM, and four Bayesian inference techniques including Bayesian Deep Ensembles (Lakshminarayanan et al., 2017), BayesTune (Kim & Hospedales, 2023), Sharpness-Aware Bayesian Neural Network (SA-BNN) (Nguyen et al., 2023a), Sharpness-aware Joint Energy-based Model (SADA-JEM) (Yang et al., 2023), Stochastic Gradient Langevin Dynamics (SGLD) (Welling & Teh, 2011), and Stein Variational Gradient Descent (SVGD) (Liu & Wang, 2016b).

We used ten warm-up epochs, batch size 64, the Gaussian kernel, and the cosine annealing learning rate scheduler for all settings. The experiments were run with PyTorch on a Tesla V100 GPU with 40GB of RAM. FHBI involves three hyperparameters: the learning rate ϵ, ascent step size ρ, and kernel width σ. We tuned these hyperparameters using the provided validation set, where the candidate sets are formed as $\epsilon \in \{0.15, 1, 1.5, 2.5\}, \rho \in \{0.01, 0.03, 0.05\}, \sigma \in \{0.7, 1, 1.2\}$. Detailed chosen hyperparameters and data augmentations for each dataset are reported in Appendix C. For each experiment, we conducted five runs of FHBI and reported the mean and standard deviation. All Bayesian methods were trained with four particles on the same set of LoRA parameters.

Experimental Results. We first present the classification accuracy results in Table 1. FHBI notably improves compared to the baselines, outperforming them in most settings. Compared to other particle sampling methods, including SGLD and SVGD, FHBI consistently performs better across all settings. Moreover, FHBI improves upon SAM by a margin of 3.2%, highlighting the advantages of using multiple particles with the underlying interactive gradient directions as previously discussed in Section 4.1. Additionally, as illustrated in Figure 2, FHBI shows the highest performance across all three domains, further solidifying its advantage over the Bayesian inference baselines.

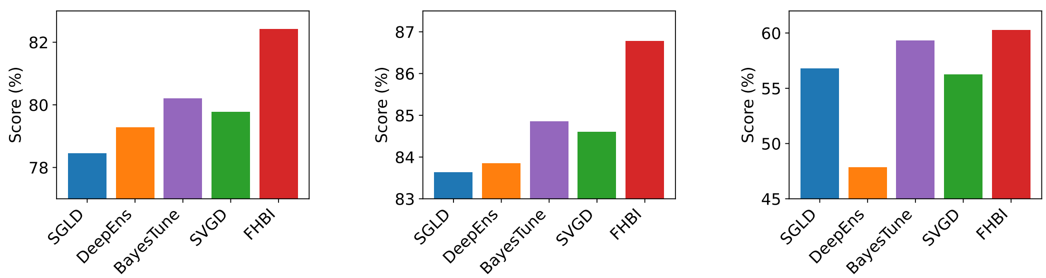


Figure 2: Domain-wise average scores on Natural (left), Specialized (middle), and Structured (right) datasets. FHBI performs best in all three domains compared to the Bayesian inference baselines.

To further assess the robustness of FHBI, we evaluate the Expected Calibration Error (ECE) of each setting. This score measures the maximum discrepancy between the model's accuracy and confidence. As indicated in Table 2, even though there is typically a trade-off between accuracy and ECE, our approach achieves a good balance between the ECE and the classification accuracy.

	Natural							Specialized				Structured								
Method	CIFAR100	Caltech101	DTD	Flower102	Pets	SVHN	Sun397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	KITTI	dSpr-Loc	dSpr-Ori	sNORB-Azi	sNORB-Ele	**AVG**
FFT	0.29	0.23	0.20	0.13	0.27	0.19	0.45	0.21	0.13	0.18	0.17	0.41	0.44	0.42	0.22	**0.14**	0.23	0.24	0.40	0.26
AdamW	0.38	0.19	0.18	0.05	0.09	0.10	0.14	0.11	0.09	0.12	0.11	**0.12**	0.19	0.34	0.18	0.14	0.21	0.18	0.31	0.17
SAM	0.21	0.25	0.20	0.11	0.12	0.15	**0.14**	0.17	0.16	0.14	0.09	0.12	0.17	0.24	0.16	0.21	0.19	0.13	0.16	0.16
DeepEns	0.24	0.12	0.22	0.04	0.10	0.13	0.23	0.16	0.07	0.15	0.21	0.31	0.32	0.36	**0.13**	0.32	0.31	0.16	0.29	0.20
BayesTune	0.32	**0.08**	0.20	**0.03**	0.85	0.12	0.22	0.13	0.07	0.13	0.22	0.12	0.23	0.30	0.24	0.28	0.28	0.31	0.26	0.23
SGLD	0.26	0.20	0.17	0.05	0.18	0.14	0.23	0.18	0.09	**0.12**	0.32	0.26	0.29	0.21	0.26	0.42	0.39	**0.11**	0.24	0.22
SADA-JEM	0.22	0.11	0.20	0.05	0.13	0.16	0.18	0.15	0.21	0.23	0.26	0.19	0.20	0.25	0.27	0.35	0.20	0.14	0.13	0.19
SA-BNN	0.22	0.08	0.19	0.15	0.12	0.12	0.24	0.13	0.06	0.12	0.18	0.14	0.21	0.22	0.24	0.25	0.41	0.46	0.34	0.20
SVGD	0.20	0.13	0.19	0.04	0.16	0.09	0.20	0.15	0.11	0.13	0.12	0.17	0.21	0.30	0.18	0.21	0.25	0.14	0.26	0.18
FHBI	**0.19**	0.10	**0.16**	0.06	**0.06**	**0.09**	0.16	**0.09**	**0.05**	0.12	**0.08**	0.14	**0.15**	**0.21**	0.15	0.16	**0.18**	0.11	**0.07**	**0.12**

Table 2: `VTAB-1K` results evaluated on the Expected Calibration Error (ECE) metric. All methods are applied to finetune the same set of LoRA parameters on ViT-B/16 pre-trained with `ImageNet-21K` dataset.

6 Ablation Studies

6.1 Effect of #particles

To understand the impact of varying the number of particles, we conducted experiments on the seven Natural datasets, reporting both accuracy and per-epoch runtime. We compared FHBI with SVGD and SAM. Figure 3 and Table 3 indicate that multiple particles result in significant performance improvements compared to a single particle. However, while increasing the number of particles enhances performance, it introduces a tradeoff regarding runtime and memory required to store the models. Based on these observations, we found that using #particles $= 4$ provides an optimal balance between performance gains and computational overhead.

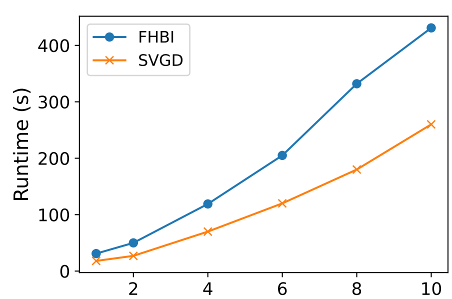


Figure 3: Runtime by #particles.

#Particles	CIFAR100	Caltech101	DTD	Flower102	Pets	SVHN	Sun397
1p (SAM)	72.7	90.3	71.4	99.0	90.2	84.4	52.4
4p	74.8	93.0	74.3	**99.4**	92.4	87.5	56.5
10p	**75.0**	**93.2**	**75.0**	99.1	**92.4**	**87.9**	**58.3**

Table 3: Accuracy by #particles.

6.2 PARTICLES SHARPNESS AND GRADIENT DIVERSITY

As discussed in Section 4.1 and Section 5, FHBI shares implicit connections with SAM by minimizing particle-wise sharpness and diversifying particle travel directions, improving the final performance. To empirically verify this hypothesis about the behavior of our algorithm, we contrast FHBI with SVGD on the KITTI dataset. Four particles are initialized at the same location. We measured: **1)** the evolution of sharpness of each particle, defined as $\max_{\|\varepsilon\| \le \rho} \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta} + \varepsilon) - \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$ according to Foret et al. (2021), and **2)** the evolution of gradients angular diversity, quantified as the Frobenius norm of the covariance matrix formed by the particle gradients. As shown in Figure 5, FHBI not only results in significantly lower and more stable sharpness evolution but also encourages less congruent gradient directions, promoting particles to explore diverse trajectories. Hence, FHBI effectively reduces particle sharpness while promoting angular diversity, improves generalization ability and avoids overfitting by collapsing into a single mode.

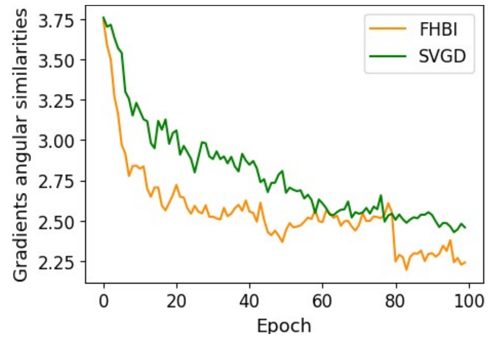
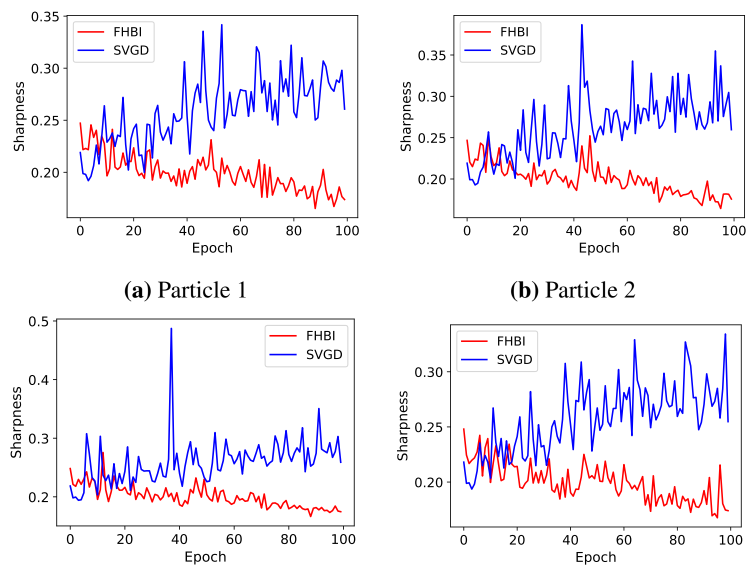


Figure 4: Gradients angular similarities with $m = 4$. Lower values indicates greater angular diversity.



(a) Particle 1 **(b)** Particle 2

(c) Particle 3 **(d)** Particle 4

Figure 5: Evolution of sharpness of particles over 100 epochs with SVGD (blue) or FHBI (red)

7 CONCLUSION

We introduce Flat Hilbert Bayesian Inference (FHBI), a particle-sampling method designed to enhance generalization ability beyond previous Bayesian inference approaches. This algorithm is based on a theoretical framework that extends generalization principles from Euclidean spaces to the infinite-dimensional RKHS. In our experiments on the VTAB-1K benchmark, FHBI consistently demonstrated performance improvements over six baseline methods by notable margins.

Limitations and Future Directions. Similar to other particle-sampling methods, FHBI needs to store multiple models. Although it remains well-suited for fine-tuning since the additional modules are typically lightweight, this requirement is a memory bottleneck for larger models. Given that the variational inference (VI) approaches can alleviate this issue, an avenue for future research is to extend the concept of *sharpness over functional spaces* introduced by our theorems to the VI techniques to improve the generalization ability of these methods without storing multiple models.

Reproducibility Statement

We open-source our implementation and provide the configs and log files at https://anonymous.4open.science/r/Flat-Hilbert-Variational-Inference-008F/

References

Momin Abbas, Quan Xiao, Lisha Chen, Pin-Yu Chen, and Tianyi Chen. Sharp-maml: Sharpness-aware model-agnostic meta learning. *arXiv preprint arXiv:2206.03996*, 2022.

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

Dara Bahri, Hossein Mobahi, and Yi Tay. Sharpness-aware minimization improves language model generalization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7360–7371, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.508. URL https://aclanthology.org/2022.acl-long.508.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Debora Caldarola, Barbara Caputo, and Marco Ciccone. Improving generalization in federated learning by seeking flat minima. In *European Conference on Computer Vision*, pp. 654–672. Springer, 2022.

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418, 2021.

Pratik Chaudhari, Anna Choromańska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer T. Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019, 2017.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *International conference on machine learning*, pp. 1683–1691. PMLR, 2014.

Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. *arXiv preprint arXiv:2106.01548*, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pp. 1019–1028. PMLR, 2017.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

Michael Dusenberry, Ghassen Jerfel, Yeming Wen, Yian Ma, Jasper Snoek, Katherine Heller, Balaji Lakshminarayanan, and Dustin Tran. Efficient and scalable Bayesian neural nets with rank-1 factors. In *International conference on machine learning*, pp. 2782–2792. PMLR, 2020.

Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *UAI*. AUAI Press, 2017.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM.

Stanislav Fort and Surya Ganguli. Emergent properties of the local geometry of neural loss landscapes. *arXiv preprint arXiv:1910.05929*, 2019.

Soumya Ghosh, Jiayu Yao, and Finale Doshi-Velez. Structured variational learning of Bayesian neural networks with horseshoe priors. In *International Conference on Machine Learning*, pp. 1744–1753. PMLR, 2018.

Alex Graves. Practical variational inference for neural networks. *Advances in Neural Information Processing Systems*, 24, 2011.

Arjun K Gupta and Daya K Nagar. *Matrix variate distributions*. Chapman and Hall/CRC, 2018.

Sepp Hochreiter and Jürgen Schmidhuber. Simplifying neural nets by discovering flat minima. In *NIPS*, pp. 529–536. MIT Press, 1994.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp, 2019. URL `https://arxiv.org/abs/1902.00751`.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL `https://arxiv.org/abs/2106.09685`.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *UAI*, pp. 876–885. AUAI Press, 2018.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *ICLR*. OpenReview.net, 2020.

Jean Kaddour, Linqing Liu, Ricardo Silva, and Matt J. Kusner. Questions for flat-minima optimization of modern neural networks. *CoRR*, abs/2202.00661, 2022. URL `https://arxiv.org/abs/2202.00661`.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *CoRR*, abs/1609.04836, 2016. URL `http://arxiv.org/abs/1609.04836`.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR*. OpenReview.net, 2017.

Minyoung Kim and Timothy M Hospedales. Bayestune: Bayesian sparse deep model fine-tuning. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, volume 36, pp. 65317–65365. Curran Associates Inc, December 2023. URL `https://neurips.cc/Conferences/2023`. Thirty-Seventh Conference on Neural Information Processing Systems, NeurIPS 2023 ; Conference date: 10-12-2023 Through 16-12-2023.

Minyoung Kim, Da Li, Shell Xu Hu, and Timothy M. Hospedales. Fisher SAM: Information geometry and sharpness aware minimisation, 2022. URL `https://arxiv.org/abs/2206.04920`.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *Advances in Neural Information Processing Systems*, 28, 2015.

Erwin Kreyszig. *Introductory Functional Analysis with Applications*. John Wiley & Sons, 1978.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017. URL `https://arxiv.org/abs/1612.01474`.

Zhouzi Li, Zixuan Wang, and Jian Li. Analyzing sharpness along gd trajectory: Progressive sharpening and edge of stability, 2022.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. *Advances in Neural Information Processing Systems*, 29, 2016a.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016b. URL `https://proceedings.neurips.cc/paper_files/paper/2016/file/b3ba8f1bee1238a2f37603d90b58898d-Paper.pdf`.

Christos Louizos and Max Welling. Multiplicative normalizing flows for variational Bayesian neural networks. In *International Conference on Machine Learning*, pp. 2218–2227. PMLR, 2017.

Thomas Möllenhoff and Mohammad Emtiyaz Khan. SAM as an optimal relaxation of bayes. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=k4fevFqSQcX`.

Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, Berlin, Heidelberg, 1996. ISBN 0387947248.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in Neural Information Processing Systems*, 30, 2017.

Van-Anh Nguyen, Tung-Long Vuong, Hoang Phan, Thanh-Toan Do, Dinh Phung, and Trung Le. Flat seeking bayesian neural networks. *Advances in Neural Information Processing Systems*, 2023a.

Van-Anh Nguyen, Tung-Long Vuong, Hoang Phan, Thanh-Toan Do, Dinh Phung, and Trung Le. Flat seeking Bayesian neural networks. In *Advances in Neural Information Processing Systems*, 2023b.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. Regularizing neural networks by penalizing confident output distributions. In *ICLR (Workshop)*. OpenReview.net, 2017.

Henning Petzka, Michael Kamp, Linara Adilova, Cristian Sminchisescu, and Mario Boley. Relative flatness and generalization. In *NeurIPS*, pp. 18420–18432, 2021.

Zhe Qu, Xingyu Li, Rui Duan, Yao Liu, Bo Tang, and Zhuo Lu. Generalized federated learning via sharpness aware minimization. *arXiv preprint arXiv:2206.02618*, 2022.

Hippolyt Ritter, Aleksandar Botev, and David 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.

Simone Rossi, Sebastien Marmin, and Maurizio Filippone. Walsh-Hadamard variational inference for Bayesian deep learning. *Advances in Neural Information Processing Systems*, 33:9674–9686, 2020.

Pengfei Wang, Zhaoxiang Zhang, Zhen Lei, and Lei Zhang. Sharpness-aware gradient matching for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3769–3778, 2023.

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 681–688, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

Xiulong Yang, Qing Su, and Shihao Ji. Towards bridging the performance gaps of joint energy-based models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15732–15741, June 2023.

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A large-scale study of representation learning with the visual task adaptation benchmark, 2020. URL `https://arxiv.org/abs/1910.04867`.

Guodong Zhang, Shengyang Sun, David Duvenaud, and Roger Grosse. Noisy natural gradient as variational inference. In *International Conference on Machine Learning*, pp. 5852–5861. PMLR, 2018.

Xingxuan Zhang, Renzhe Xu, Han Yu, Yancheng Dong, Pengfei Tian, and Peng Cui. Flatness-aware minimization for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5189–5202, 2023.

Tian-Yi Zhou, Namjoon Suh, Guang Cheng, and Xiaoming Huo. Approximation of RKHS functionals by neural networks, 2024. URL `https://arxiv.org/abs/2403.12187`.

SUPPLEMENT TO "IMPROVING GENERALIZATION WITH FLAT HILBERT VARIATIONAL INFERENCE"

A MISSING PROOFS

We introduce a few additional notations for the sake of the missing proofs of the main theoretical results. Given a RKHS $\mathcal{H}$ equipped with the inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ and the norm operator $\|\cdot\|_{\mathcal{H}}$. We define the single-sample loss function on the functional space $\mathcal{H}$ to be a map:

$$\begin{aligned}\tilde{\ell}:\mathcal{H}^d\times\mathcal{X}\times\mathcal{Y}&\to\mathbb{R}\\ (\boldsymbol{f},x,y)&\mapsto\tilde{\ell}(\boldsymbol{f},(x,y)).\end{aligned}$$

Define the *general functional loss* $\tilde{L}_{\mathcal{D}}(\boldsymbol{f})=\mathbb{E}_{(x,y)\sim\mathcal{D}}[\tilde{\ell}(\boldsymbol{f},(x,y))]$ and the *empirical functional loss* $\tilde{L}_{\mathcal{S}}(\boldsymbol{f})=\sum_{i=1}^{n}\tilde{\ell}(\boldsymbol{f},(x_i,y_i))$. Throughout the proof, we assume that the parameter space is bounded by $\|\boldsymbol{\theta}\|\le H$, and the data is al bounded that $\|x\|\le R, y\le R$ for some $R,H\in\mathbb{R}$.

We introduce the following lemmas that will be used throughout the proof of our main theorems.

Lemma 2 (Approximation of RKHS functionals)**.** *Let $d\in\mathbb{N},\mathcal{X}=[-R,R]^K$ for some $K\in\mathbb{R}$. Consider $\mathcal{K}=\{f\in\mathcal{H}:\|f\|_{\mathcal{H}}\le 1\}$ with $\mathcal{H}$ induced by some Mercer kernel which is α-Holder continuous for $\alpha\in(0,1)$ with constant $C_K\ge 0$. Suppose F is $s-$Holder continuous for $s\in(0,1]$ with constant $C_f\ge 0$. There exists some $M_0\in\mathbb{N}$ such that for every $M\in\mathbb{N}$ with $M>M_0$, by taking some fixed $\bar{t}=\{t_i\}$ with $N\in\mathbb{N}$, we have a tanh neural network $\hat{G}$ with two hidden layers of widths at most $N(M-1)$ and $3\frac{N+1}{2}(5M)^N$ parameters satisfying*

$$\sup_{f\in\mathcal{K}}|F(f)-\hat{G}(f(\bar{t}))|\le RC_F(\epsilon_K(\bar{t}))^s+\frac{7N^2RC_G}{M}, \tag{18}$$

with

$$C_G=C_F(1+\|K[\bar{t}]^{-1}\|_{op}\sqrt{N}C_K(h_{\bar{t}})^\alpha)^s,$$

where $K[\bar{t}]$ is the Gram matrix of $\bar{t}$.

Proof. The proof can be found in Zhou et al. (2024) □

Lemma 3 (Product of RKHSs)**.** *Given n RKHSs $\mathcal{H}_1,\mathcal{H}_2,\cdots,\mathcal{H}_n$, each defined on corresponding sets $X_1,X_2,\cdots,X_n$ with kernels $k_1(x_1,y_1),\cdots k_n(x_n,y_n)$ respectively. Then, $\mathcal{H}=\bigotimes_{i=1}^n=\mathcal{H}_1\times\mathcal{H}_2\times\cdots\times\mathcal{H}_n$ is also an RKHS, with kernel K that is the product of the individual kernels.*

Proof. The product space $\mathcal{H}=\bigotimes_{i=1}^n\mathcal{H}_i$ consists of tuples of functions $(f_1,f_2,\cdots,f_n)$. Firstly, we define the inner product in $\mathcal{H}$ as:

$$\langle(f_1,f_2,\cdots,f_n),(g_1,g_2,\cdots,g_n)\rangle_{\mathcal{H}}=\sum_{i=1}^n\langle f_i,g_i\rangle_{\mathcal{H}_i}.$$

This definition naturally defines a Hilbert space structure on $\mathcal{H}$ since each $\mathcal{H}_i$ is a Hilbert space, and the sum of inner products is linear and positive definite. Now we define the kernel for the product space:

$$k((x_1,x_2,\cdots,x_n),(y_1,y_2,\cdots,y_n))=\prod_{i=1}^n k(x_i,y_i).$$

Notice that the pointwise product of positive definite kernels is a positive definite kernel, hence this kernel is valid.

We now verify the reproducing property of $\mathcal{H}$. Consider a function $f=(f_1,f_2,\cdots,f_n)\in\mathcal{H}$, and evaluate the function at a point $(x_1,x_2,\cdots,x_n)\in\bigotimes_{i=1}^n X_i$.

The reproducing property in each individual RKHS $\mathcal{H}_i$ implies that:

$$f_i(x_i)=\langle f_i,k_i(x_i,\cdot)\rangle_{\mathcal{H}_i}.$$

Hence, for the function $f = (f_1, \cdots, f_n)$, we get:

$$\begin{aligned} f((x_1, x_2, \cdots, x_n)) &= (f_1(x_1), f_2(x_2), \cdots, f_n(x_n)) \\ &= (\langle f_1, k_1(x_1, \cdot)\rangle_{\mathcal{H}_1}, \langle f_2, k_2(x_2, \cdot)\rangle_{\mathcal{H}_2}, \cdots, \langle f_n, k_n(x_n, \cdot)\rangle_{\mathcal{H}_n}) \\ &= \langle (f_1, f_2, \cdots, f_n), (k_1(x_1, \cdot), k_2(x_2, \cdot), \cdots k_n(x_n, \cdot))\rangle_{\mathcal{H}}. \end{aligned}$$

Thus, the reproducing property holds for the product space $\mathcal{H}$. Since $\mathcal{H}$ is a Hilbert space and the kernel k satisfies the reproducing property, we conclude that $\mathcal{H} = \bigotimes_{i=1}^n \mathcal{H}_i$ is another RKHS. $\square$

A.1 Proof of Proposition 1

Proposition 1. *Consider the problem of finding the distribution $\mathbb{Q}$ that solves:*

$$\mathbb{Q}^* = \min_{\mathbb{Q} \ll \mathbb{P}_\theta} \left\{ \mathbb{E}_{\theta \sim \mathbb{Q}}[\mathcal{L}_\mathcal{D}(\boldsymbol{\theta})] + D_{\mathrm{KL}}(\mathbb{Q} \| \mathbb{P}_\theta) \right\} \tag{19}$$

where we search over $\mathbb{Q}$ absolutely continuos w.r.t $\mathbb{P}_{\boldsymbol{\theta}}$, and the second term is the regularization term. The closed-form solution to this problem is the ***population posterior*** *whose density has the form:*

$$q^*(\boldsymbol{\theta}) \propto \exp(-\mathcal{L}_\mathcal{D}(\boldsymbol{\theta}))p(\boldsymbol{\theta}).$$

Proof. This proposition is the general case of **Theorem 3.1** by Nguyen et al. (2023b). Denote $q(\cdot)$ as the density function of $\mathbb{Q}$. We have:

$$\mathbb{E}_{\theta \sim \mathbb{Q}}[\mathcal{L}_\mathcal{D}(\boldsymbol{\theta})] + D_{\mathrm{KL}}(\mathbb{Q} \| \mathbb{P}_\theta) = \int_\Theta \mathcal{L}_\mathcal{D}(\boldsymbol{\theta}) q(\boldsymbol{\theta}) d\boldsymbol{\theta} + \int_\Theta q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})} d\boldsymbol{\theta}.$$

The Lagrangian is given by:

$$L(q, \alpha) = \int_\Theta \mathcal{L}_\mathcal{D}(\boldsymbol{\theta}) q(\boldsymbol{\theta}) d\boldsymbol{\theta} + \int_\Theta q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})} d\boldsymbol{\theta} + \alpha\left(\int q(\boldsymbol{\theta}) d\boldsymbol{\theta} - 1\right).$$

Taking derivative with respect to $q(\boldsymbol{\theta})$, it follows

$$\begin{aligned} &\mathcal{L}_\mathcal{D} + \log q(\boldsymbol{\theta}) + 1 - \log p(\boldsymbol{\theta}) + \alpha = 0, \\ &q(\boldsymbol{\theta}) = \exp(-\mathcal{L}_\mathcal{D}(\theta)) p(\boldsymbol{\theta}) \exp(-\alpha - 1), \end{aligned}$$

which implies that

$$q(\boldsymbol{\theta}) \propto \exp(-\mathcal{L}_\mathcal{D}(\boldsymbol{\theta}))p(\boldsymbol{\theta}).$$

Then, the optimal solution is the population posterior $p(\boldsymbol{\theta}|\mathcal{S})$, which concludes the proof. $\square$

A.2 Proof of Theorem 1

Theorem 1. *For any $\rho > 0$ and any distribution $\mathcal{D}$, with probability $1 - \delta$ over the choice of the training set $\mathcal{S} \sim \mathcal{D}^n$,*

$$\tilde{L}_\mathcal{D}(\boldsymbol{f}) \le \max_{\|\boldsymbol{f}' - \boldsymbol{f}\|_{\mathcal{H}^d} \le \rho} \tilde{L}_\mathcal{S}(\boldsymbol{f}') + \sqrt{\frac{N' \log\left(1 + \frac{C}{\rho^2 P^2}\left(1 + \sqrt{\frac{\log(N)}{N'}}\right)^2\right) + 4\log\frac{n}{\delta} + 8\log(6n + 3k)}{n - 1}}.$$

Proof. $\tilde{\ell}$ is a functional that maps from $\mathcal{H}^d \times \mathcal{X} \times \mathcal{Y}$ to $\mathbb{R}$. Notice that $\mathcal{H}^d$ is a RKHS, $\mathcal{X} = \mathbb{R}^a$ and $\mathcal{Y} = \mathbb{R}^b$ for some $a, b \in \mathbb{Z}$ are Euclidean spaces, which are also instances of RKHS. Moreover, the product of RKHS's is also a RKHS according to Lemma 3. Hence, $\mathcal{H}^d \times \mathcal{X} \times \mathcal{Y}$ is also a RKHS.

According to Lemma 2, there exists N points $\overline{\boldsymbol{\theta}} = \{\boldsymbol{\theta}_i\}_{i=1}^{N} \subset \Theta$, and a two-layer neural network $G_{\boldsymbol{W}}$ parameterized by $\boldsymbol{W}$ so that

$$|\tilde{\ell}(\boldsymbol{f}, x, y) - G_{\boldsymbol{W}}(\boldsymbol{f}(\overline{\boldsymbol{\theta}}), x, y)| \leq RC_F(\epsilon_K(\bar{t}))^s + \frac{7N^2 RC_G}{M},$$

for every $(\boldsymbol{f}, x, y) \in \mathcal{H}^d \times \mathcal{X} \times \mathcal{Y}$. Consider $\boldsymbol{f}' \in \mathcal{H}^d$ so that $\|\boldsymbol{f}' - \boldsymbol{f}\| \leq \rho$, it implies $|\boldsymbol{f}(\overline{\boldsymbol{\theta}}) - \boldsymbol{f}'(\overline{\boldsymbol{\theta}})| \leq P\|\boldsymbol{f} - \boldsymbol{f}'\|_{\mathcal{H}^d} \leq P\rho$. Denote $\tilde{\boldsymbol{\theta}} = \boldsymbol{f}(\overline{\boldsymbol{\theta}}) \in \mathbb{R}^{N'}$ for some $N' \in \mathbb{Z}$, by invoking the inequality from Foret et al. (2021), let $\rho' = \rho P$, it follows that:

$$\begin{aligned}
\tilde{L}_{\mathcal{D}}(\boldsymbol{f}) &= \mathbb{E}_{(x,y)\sim\mathcal{D}}[\tilde{\ell}(\boldsymbol{f}, x, y)] \leq \mathbb{E}_{(x,y)\sim\mathcal{D}}[G_{\boldsymbol{W}}(\boldsymbol{f}(\overline{\boldsymbol{\theta}}), x, y)] + RC_F(\epsilon_K(\bar{t}))^s + \frac{7N^2 RC_G}{M} \\
&= \mathbb{E}_{(x,y)\sim\mathcal{D}}[G_{\boldsymbol{W}}(\tilde{\boldsymbol{\theta}}, x, y)] + RC_F(\epsilon_K(\bar{t}))^s + \frac{7N^2 RC_G}{M} \\
&\leq \max_{\|\tilde{\boldsymbol{\theta}}' - \tilde{\boldsymbol{\theta}}\|_2^2 \leq \rho'} \frac{1}{n} \sum_{i=1}^{n} G_{\boldsymbol{W}}(\tilde{\boldsymbol{\theta}}', x, y) + h(M, N) \\
&+ \sqrt{\frac{N' \log\left(1 + \frac{C}{\rho'^2}\left(1 + \sqrt{\frac{\log(N)}{N'}}\right)^2\right) + 4\log\frac{n}{\delta} + 8\log(6n + 3k)}{n-1}}.
\end{aligned}$$

By definition, a RKHS is a closed Hilbert space. Then, there exists a sequence $\{\boldsymbol{f}'_n\}$ so that $\boldsymbol{f}'_n(\overline{\boldsymbol{\theta}})$ that gets arbitrarily close to $\tilde{\boldsymbol{\theta}}'$. Then, for any $\epsilon > 0$, it follows:

$$\begin{aligned}
\tilde{L}_{\mathcal{D}}(\boldsymbol{f}) &\leq \max_{\|\tilde{\boldsymbol{\theta}}' - \tilde{\boldsymbol{\theta}}\|_2^2 \leq \rho'} \frac{1}{n} \sum_{i=1}^{n} G_{\boldsymbol{W}}(\tilde{\boldsymbol{\theta}}', x, y) + h(M, N) \\
&\quad + \sqrt{\frac{N' \log\left(1 + \frac{C}{\rho'^2}\left(1 + \sqrt{\frac{\log(N)}{N'}}\right)^2\right) + 4\log\frac{n}{\delta} + 8\log(6n + 3k)}{n-1}} \\
&\quad \leq \max_{\|\boldsymbol{f}'(\overline{\boldsymbol{\theta}}) - \boldsymbol{f}(\overline{\boldsymbol{\theta}})\|_2^2 \leq \rho P} \frac{1}{n} \sum_{i=1}^{n} G_{\boldsymbol{W}}(\boldsymbol{f}'(\overline{\boldsymbol{\theta}}), x, y) + h(M, N) + \epsilon\mathcal{O}(1) \\
&\quad + \sqrt{\frac{N' \log\left(1 + \frac{C}{\rho^2 P^2}\left(1 + \sqrt{\frac{\log(N)}{N'}}\right)^2\right) + 4\log\frac{n}{\delta} + 8\log(6n + 3k)}{n-1}} \\
&\quad \leq \max_{\|\boldsymbol{f}' - \boldsymbol{f}\|_2^2 \leq \rho} \frac{1}{n} \sum_{i=1}^{n} G_{\boldsymbol{W}}(\boldsymbol{f}'(\boldsymbol{\theta}), x, y) + h(M, N) + \epsilon\mathcal{O}(1) \\
&\quad + \sqrt{\frac{N' \log\left(1 + \frac{C}{\rho^2 P^2}\left(1 + \sqrt{\frac{\log(N)}{N'}}\right)^2\right) + 4\log\frac{n}{\delta} + 8\log(6n + 3k)}{n-1}} \\
&\quad \leq \max_{\|\boldsymbol{f}' - \boldsymbol{f}\|_2^2 \leq \rho} \tilde{L}_{\mathcal{S}}(\boldsymbol{f}') + h(M, N) + \epsilon\mathcal{O}(1) \\
&\quad + \sqrt{\frac{N' \log\left(1 + \frac{C}{\rho^2 P^2}\left(1 + \sqrt{\frac{\log(N)}{N'}}\right)^2\right) + 4\log\frac{n}{\delta} + 8\log(6n + 3k)}{n-1}}.
\end{aligned}$$

This is true for any $\epsilon > 0$. Moreover, we can choose ϵ_K and M to be arbitrarily small so that $h(M, N) \to 0$. Hence, it implies

$$\tilde{L}_{\mathcal{D}}(\boldsymbol{f}) \leq \max_{\|\boldsymbol{f}'-\boldsymbol{f}\|_2^2 \leq \rho} \tilde{L}_{\mathcal{S}}(\boldsymbol{f}')+ \\ + \sqrt{\frac{N' \log\left(1 + \frac{C}{\rho^2 P^2}\left(1 + \sqrt{\frac{\log(N)}{N'}}\right)^2\right) + 4\log\frac{n}{\delta} + 8\log(6n+3k)}{n-1}},$$

which concludes our proof. □

A.3 Proof of Theorem 2

Now we can prove the Theorem 2. We restate the theorem

Theorem 2. *For any target distribution p, reference distribution q, and any $\rho > 0$, we have the following bound between the general KL loss and the empirical KL loss*

$$D_{\mathrm{KL}}\left(q_{[\boldsymbol{f}]}||p\Big(\boldsymbol{\theta}|\mathcal{D}\Big)\right) \leq \max_{\boldsymbol{f}' \in \mathcal{B}_\rho(\boldsymbol{f})} D_{\mathrm{KL}}\left(q_{[\boldsymbol{f}']}||p\Big(\boldsymbol{\theta}|\mathcal{S}\Big)\right) \\ + \sqrt{\frac{N' \log\left(1 + \frac{C}{\rho^2 P^2}\left(1 + \sqrt{\frac{\log(N)}{N'}}\right)^2\right) + 4\log\frac{n}{\delta} + 8\log(6n+3k)}{n-1}}.$$

Proof. Consider the left-hand side, we have:

$$\begin{aligned} D_{\mathrm{KL}}(q_{[\boldsymbol{f}]}\|p(\boldsymbol{\theta}|\mathcal{D})) &= \int q_{[\boldsymbol{f}]}(\boldsymbol{\theta})\left(\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}) + \log\frac{q_{[\boldsymbol{f}]}(\boldsymbol{\theta})}{p(\boldsymbol{\theta})} + \log Z_{\mathcal{D}}\right)d\boldsymbol{\theta} \\ &= \int q_{[\boldsymbol{f}]}(\boldsymbol{\theta})\left(\mathbb{E}_{(x,y)\sim\mathcal{D}}\ell(\boldsymbol{\theta}, x, y) + \log\frac{q_{[\boldsymbol{f}]}(\boldsymbol{\theta})}{p(\boldsymbol{\theta})} + \log Z_{\mathcal{D}}\right)d\boldsymbol{\theta} \\ &= \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\int q_{[\boldsymbol{f}]}(\boldsymbol{\theta})\Big(\ell(\theta; x, y) + \log\frac{q_{[\boldsymbol{f}]}(\boldsymbol{\theta})}{p(\boldsymbol{\theta})}\Big)d\boldsymbol{\theta}\right] + \int q_{[\boldsymbol{f}]}(\boldsymbol{\theta})\log Z_{\mathcal{D}} d\boldsymbol{\theta}. \end{aligned}$$

On the other hand, we also have:

$$\begin{aligned} D_{\mathrm{KL}}(q_{[\boldsymbol{f}]}\|p(\boldsymbol{\theta}|\mathcal{S})) &= \int q_{[\boldsymbol{f}]}\left((\boldsymbol{\theta})\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}) + \log\frac{q_{[\boldsymbol{f}]}(\boldsymbol{\theta})}{p(\boldsymbol{\theta})} + \log Z_{\mathcal{S}}\right)d\boldsymbol{\theta} \\ &= \int q_{[\boldsymbol{f}]}(\boldsymbol{\theta})\left(\frac{1}{n}\sum_{i=1}^{n}\ell(\boldsymbol{\theta}, x_i, y_i) + \log\frac{q_{[\boldsymbol{f}]}(\boldsymbol{\theta})}{p(\boldsymbol{\theta})} + \log Z_{\mathcal{S}}\right)d\boldsymbol{\theta} \\ &= \frac{1}{n}\sum_{i=1}^{n}\left[\int q_{[\boldsymbol{f}]}(\boldsymbol{\theta})\Big(\ell(\theta; x_i, y_i) + \log\frac{q_{[\boldsymbol{f}]}(\boldsymbol{\theta})}{p(\boldsymbol{\theta})} + \log Z_{\mathcal{S}}\Big)d\boldsymbol{\theta}\right] + \int q_{[\boldsymbol{f}]}(\boldsymbol{\theta})\log Z_{\mathcal{S}} d\boldsymbol{\theta}. \end{aligned}$$

We define $\tilde{L}$ to be the functional such that:

$$\begin{aligned} &\tilde{L} : \mathcal{H}^d \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R} \\ &\qquad (\boldsymbol{f}, x, y) \mapsto \tilde{L}(\boldsymbol{f}, x, y) = \int q_{[\boldsymbol{f}]}(\boldsymbol{\theta})\Big(\ell(\theta; x, y) + \log\frac{q_{[\boldsymbol{f}]}(\boldsymbol{\theta})}{p(\boldsymbol{\theta})}\Big)d\boldsymbol{\theta}. \end{aligned}$$

According to Theorem 1, we have:

$$\tilde{L}_{\mathcal{D}}(\boldsymbol{f}) \leq \max_{\|\boldsymbol{f}'-\boldsymbol{f}\|_{\mathcal{H}^d}\leq\rho} \tilde{L}_{\mathcal{S}}(\boldsymbol{f}') \tag{20}$$

$$+\sqrt{\frac{N'\log\left(1+\frac{C}{\rho^2P^2}\left(1+\sqrt{\frac{\log(N)}{N'}}\right)^2\right)+4\log\frac{n}{\delta}+8\log(6n+3k)}{n-1}}. \tag{21}$$

Moreover, the model and data spaces are bounded, so $\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta})$ and $\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$ are bounded. Then, there exists constants d, D such that $d \leq \log Z_{\mathcal{S}}, \log Z_{\mathcal{D}} \leq D$, which also implies $d \leq \int q_{[\boldsymbol{f}]} \log Z_{\mathcal{S}} d\boldsymbol{\theta} \leq D$ and $d \leq \int q_{[\boldsymbol{f}]} \log Z_{\mathcal{D}} d\boldsymbol{\theta} \leq D$. It follows that for all $\boldsymbol{f}, \boldsymbol{f}' \in \mathcal{H}^d$:

$$\int q_{[\boldsymbol{f}]}(\boldsymbol{\theta}) \log Z_{\mathcal{D}} d\boldsymbol{\theta} \leq \int q_{[\boldsymbol{f}']}(\boldsymbol{\theta}) \log Z_{\mathcal{S}} d\boldsymbol{\theta} + D - d. \tag{22}$$

Combining the Inequalities 21 and 22, it follows that:

$$D_{\mathrm{KL}}\left(q_{[\boldsymbol{f}]}||p\left(\boldsymbol{\theta}|\mathcal{D}\right)\right) \leq \max_{\|\boldsymbol{f}'-\boldsymbol{f}\|_{\mathcal{H}^d}\leq\rho} D_{\mathrm{KL}}\left(q_{[\boldsymbol{f}']}||p\left(\boldsymbol{\theta}|\mathcal{S}\right)\right)$$
$$+\sqrt{\frac{N'\log\left(1+\frac{C}{\rho^2P^2}\left(1+\sqrt{\frac{\log(N)}{N'}}\right)^2\right)+4\log\frac{n}{\delta}+8\log(6n+3k)}{n-1}}.$$

which concludes our proof. □

B Additional Experiment: Effect of kernel choice

The implementation of FHBI relies on the choice of the kernel k. In our experiments, we selected the RBF kernel due to its widespread use in the kernel methods literature, known for its strong representational capabilities and its ability to balance underfitting and overfitting through the kernel width parameter σ. To evaluate the impact of different kernel choices, we tested our method on the four Specialized datasets using the polynomial kernel of degree 10 as a comparison. The results, summarized in Table 4, indicate that while the polynomial kernel slightly underperforms relative to the RBF kernel, the difference is minimal, with a performance gap of less than 0.3%.

Kernel	Camelyon	EuroSAT	Resisc45	Retinopathy	AVG
RBF	**85.3**	**95.0**	**87.2**	**79.6**	**86.8**
Polynomial (d=10)	85.0	94.9	86.8	79.2	86.5

Table 4: Classification accuracy on the Specialized datasets with different kernel choices

C Experimental Details

C.1 Chosen Hyperparameters

We grid-search hyperparameters on the validation set, where the key hyperparameters are: the kernel width σ, the initial learning rate ϵ, and the ascend step size ρ. The candidate sets are formed as $\epsilon \in \{0.15, 1, 1.5, 2.1, 2.5\}, \rho \in \{0.01, 0.03, 0.05\}, \sigma \in \{0.7, 1, 1.2\}$. The chosen hyperparameters are as follows (ϵ, ρ, σ): `CIFAR100` = $(0.15, 0.03, 1.2)$, `Caltech101` = $(2.1, 0.05, 1.2)$, `DTD` = $(0.15, 0.03, 1.2)$, `Flowers102` = $(0.15, 0.03, 1)$, `Pets` = $(0.15, 0.03, 1.2)$, `SVHN` = $(2.5, 0.01, 1)$, `Sun397` = $(0.15, 0.03, 1.2)$, `Patch-Camelyon` = $(2.1, 0.05, 1)$, `DMLab` = $(2.1, 0.03, 1)$, `EuroSAT` = $(2.5, 0.01, 1.2)$, `Resisc45` = $(1.5, 0.03, 1.2)$, `Diabetic-Retinopathy` = $(2.1, 0.03, 1)$, `Clevr-Count` = $(2.5, 0.01, 1)$, `Clevr-Dist` = $(1, 0.01, 1.2)$, `KITTI` = $(2.1, 0.05, 1)$, `dSprites-loc` = $(2.1, 0.05, 1)$, `dSprites-ori` = $(2.1, 0.03, 1.2)$, `smallNorb-azi` = $(1, 0.05, 1)$, `smallNorb-ele` = $(1, 0.03, 0.7)$.

C.2 Data Augmentations

Our implementation is based on the repository V-PETL. Similar to this repository, we use a different data augmentation among the following three augmentations for each dataset. In particular, the data augmentations that we used for each setting are:

- For `CIFAR100, DTD, Flower102, Pets, Sun397`

```
self.transforms_train = transforms.Compose(
    [
        transforms.RandomResizedCrop(
            (self.size, self.size),
            scale=(self.min_scale, self.max_scale),
        ),
        transforms.RandomHorizontalFlip(self.flip_prob),
        transforms.TrivialAugmentWide()
        if self.use_trivial_aug
        else transforms.RandAugment(self.rand_aug_n,
                                    self.rand_aug_m),
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.485, 0.456, 0.406],
                             std=[0.229, 0.224, 0.225])]),
        transforms.RandomErasing(p=self.erase_prob),
    ]
)
self.transforms_test = transforms.Compose(
    [
        transforms.Resize(
            (self.size, self.size),
        ),
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.485, 0.456, 0.406],
                             std=[0.229, 0.224, 0.225])]),
    ]
)
```

- For `Caltech101, Clevr-Dist, Dsprites-Loc, Dsprites-Ori, SmallNorb-Azi, SmallNorb-Ele`:

```
self.transform_train = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406],
                         std=[0.229, 0.224, 0.225])])
self.transform_test = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406],
                         std=[0.229, 0.224, 0.225])])
```

- For `Clevr-Count, DMLab, EuroSAT, KITTI, Patch Camelyon, Resisc45, SVHN, Diabetic Retinopathy`:

```
from timm.data import create_transform
self.transform_train = create_transform(
            input_size=(224, 224),
            is_training=True,
            color_jitter=0.4,
            auto_augment='rand-m9-mstd0.5-inc1',
            re_prob=0.0,
            re_mode='pixel',
```

```
                re_count=1,
                interpolation='bicubic',
            )
        aug_transform.transforms[0] = transforms.Resize((224, 224),
                                                         interpolation=3)
        self.transform_test = transforms.Compose([
                transforms.Resize((224, 224)),
                transforms.ToTensor(),
                transforms.Normalize(mean=[0.485, 0.456, 0.406],
                                     std=[0.229, 0.224, 0.225])])
```