Repulsive Deep Ensembles are Bayesian

Anonymous Author(s)
Affiliation
Address
email

Abstract

Deep ensembles have recently gained popularity in the deep learning community for their conceptual simplicity and efficiency. However, maintaining functional diversity between ensemble members that are independently trained with gradient descent is challenging. This can lead to pathologies when adding more ensemble members, such as a saturation of the ensemble performance, which converges to the performance of a single model. Moreover, this does not only affect the quality of its predictions, but even more so the uncertainty estimates of the ensemble, and thus its performance on out-of-distribution data. We hypothesize that this limitation can be overcome by discouraging different ensemble members from collapsing to the same function. To this end, we introduce a kernelized repulsive term in the update rule of the deep ensembles. We show that this simple modification not only enforces and maintains diversity among the members but, even more importantly, transforms the maximum a posteriori inference into proper Bayesian inference. Namely, we show that the training dynamics of our proposed repulsive ensembles follow a Wasserstein gradient flow of the KL divergence with the true posterior. We study repulsive terms in weight and function space and empirically compare their performance to standard ensembles and Bayesian baselines on synthetic and real-world prediction tasks.

1 Introduction

There have been many recent advances on the theoretical properties of sampling algorithms for approximate Bayesian inference, which changed our interpretation and understanding of them. Particularly worth mentioning is the work of Jordan et al. [32], who reinterpret Markov Chain Monte Carlo (MCMC) as a gradient flow of the KL divergence over the Wasserstein space of probability measures. This new formulation allowed for a deeper understanding of approximate inference methods but also inspired the inception of new and more efficient inference strategies.

Following this direction, Liu and Wang [43] recently proposed the Stein Variational Gradient Descent (SVGD) method to perform approximate Wasserstein gradient descent. Conceptually, this method, that belongs to the family of particle-optimization variational inference (POVI), introduces a repulsive force through a kernel acting in the parameter space to evolve a set of samples towards high-density regions of the target distribution without collapsing to a point estimate. However, when dealing with neural networks, the Bayesian approach is not the only one capable of offering calibrated models that allow for uncertainty estimation; ensembles of neural networks (so-called *deep ensembles*) have indeed achieved great success recently, both in terms of predictive performance [35, 64] as well as uncertainty estimation [53], and constitute the main competitor to the Bayesian counterpart. That being said, while they might allow for the averaging of predictions over several hypotheses, they do not offer any guarantees for the diversity between those hypotheses nor do they benefit of the advantages of the probabilistic Bayesian framework.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

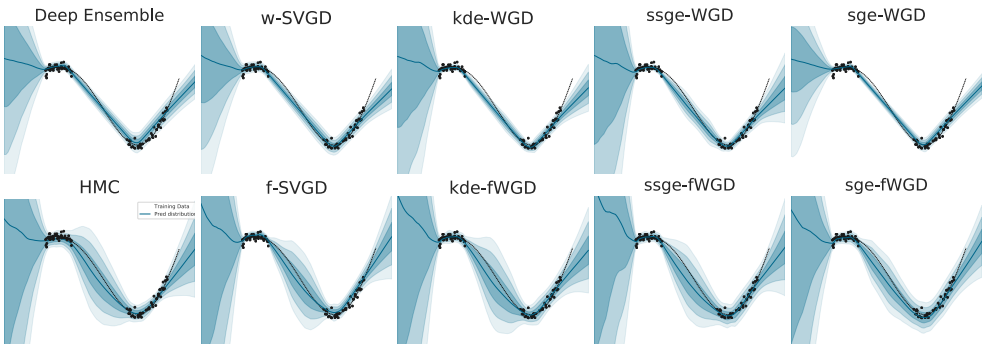

Figure 1: **BNN 1D regression.** The function-space methods (SVGD and WGD) approach the HMC posterior more closely, while the standard deep ensembles and weight-space methods fail to properly account for the uncertainty, especially the in-between uncertainty.

In this work, we show how the introduction of a repulsive term between the members in the ensemble, inspired by SVGD, not only naïvely guarantees the diversity among the members, avoiding their collapse in the parameter space, but also allows for a reformulation of the method as a gradient flow of the KL divergence in the Wasserstein space of distributions. It thus allows to reformulate deep ensembles with repulsion as a truly Bayesian method.

Nevertheless, BNN inference in weight space can lead to degenerate solutions, due to the over-parametrization of these models. That is, several samples could have very different weights but map to the same function, thus giving a false sense of diversity in the ensemble. This property, that we will refer to as *non-identifiability* of neural networks (see Appendix A), can lead to redundancies in the posterior distribution. It implies that methods like MCMC sampling, deep ensembles, and SVGD waste computation in local modes that account for equivalent functions. Predictive distributions approximated using samples from these modes do not improve over a simple point estimate and lead to a poor uncertainty estimation. Following this idea, Wang et al. [60] introduced a new method to extend POVI methods to function space, overcoming this limitation. Here, we also study an update rule that allows for an approximation of the gradient flow of the KL divergence in function space instead of weight space.

We make the following contributions:

- We derive several different repulsion terms that can be added as regularizers to the gradient updates of deep ensembles to endow them with Bayesian properties.
- We show that these terms approximate Wasserstein gradient flows of the KL divergence and can be used both in weight space and function space.
- We compare these proposed methods theoretically to standard deep ensembles and SVGD and highlight their different guarantees.
- We assess all these methods on synthetic and real-world deep learning tasks and show that our proposed repulsive ensembles can achieve competitive performance and improved uncertainty estimation.

2 Repulsive Deep Ensembles

In supervised deep learning, we typically consider a likelihood function $p(\boldsymbol{y}|f(\boldsymbol{x};\mathbf{w}))$ (e.g., Gaussian for regression or Categorical for classification) parameterized by a neural network $f(\boldsymbol{x};\boldsymbol{w})$ and training data $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$. In Bayesian neural networks (BNNs), we are interested in the posterior distribution of all likely networks given by $p(\mathbf{w}|\mathcal{D}) \propto \prod_{i=1}^n p(\boldsymbol{y}_i|f(\boldsymbol{x}_i;\mathbf{w}))\, p(\mathbf{w})$, where $p(\boldsymbol{w})$ is the prior distribution over weights. Crucially, when making a prediction on a test point $\boldsymbol{x}^*$, in the Bayesian approach we do not only use a single parameter $\hat{\mathbf{w}}$ to predict $\boldsymbol{y}^* = f(\boldsymbol{x}^*;\hat{\mathbf{w}})$, but we marginalize over the whole posterior, thus taking all possible explanations of the data into account:

$$p(\boldsymbol{y}^*|\boldsymbol{x}^*, \mathcal{D}) = \int p(\boldsymbol{y}^*|f(\boldsymbol{x}^*;\mathbf{w}))\, p(\mathbf{w}|\mathcal{D})\, \mathrm{d}\boldsymbol{w} \tag{1}$$

While approximating the posterior of deep neural networks (or sampling from it) is a challenging task, performing maximum a posteriori (MAP) estimation, which corresponds to finding the mode of the posterior, is usually simple. Ensembles of neural networks use the non-convexity of the MAP optimization problem to create a collection of K independent—and possibly different— solutions. Considering n weight configurations of neural networks $\{\mathbf{w}_i\}_{i=1}^n$ with $\mathbf{w}_i \in \mathbb{R}^d$, the dynamics of the ensemble under the gradient of the posterior lead to the following update rule at iteration t:

$$\begin{aligned} \mathbf{w}_i^{t+1} &\leftarrow \mathbf{w}_i^t + \epsilon_t \phi(\mathbf{w}_i^t) \\ \text{with} \qquad \phi(\mathbf{w}_i^t) &= \nabla_{\mathbf{w}_i^t} \log p(\mathbf{w}_i^t|\mathcal{D}) \,, \end{aligned} \tag{2}$$

with step size ϵ_t. Ensemble methods have a long history [e.g., 38, 22, 5] and were recently revisited for neural networks [35] and coined *deep ensembles*. The predictions of the different members are combined to create a predictive distribution by using the solutions to compute the Bayesian model average (BMA) in Eq. (1). Recent works [53] have shown that deep ensembles can outperform some of the Bayesian approaches for uncertainty estimation. Even more recently, Wilson and Izmailov [66] argued that deep ensembles can be considered a compelling approach to Bayesian model averaging. Despite these ideas, the ability of deep ensembles to efficiently average over multiple hypotheses and to explore the functional landscape of the posterior distribution studied in [15] does not make the method necessarily Bayesian. Indeed, the additional Langevin noise introduced in [61], which is not considered in deep ensembles, is crucial to ensure samples from the correct posterior distribution.

From a practical standpoint, since the quality of an ensemble hinges on the diversity of its members, many methods were recently proposed to improve this diversity without compromising the individual accuracy. For instance, Wenzel et al. [64] propose hyper-deep ensembles that combine deep networks with different hyperparameters. Similarly, cyclical learning-rate schedules can explore several local minima for the ensemble members [27]. Alternatively, Rame and Cord [54] proposed an information-theoretic framework to avoid redundancy in the members. However, the absence of a constraint that prevents particles from converging to the same mode limits the possibility of improvement by introducing more ensemble members. This means that any guarantees to converge to different modes must exclusively rely on:

1. The randomness of the initialization.
2. The noise in the estimation of the gradients due to minibatching.
3. The number of local optima that might be reached during gradient descent.

Moreover, the recent study of Geiger et al. [21] showed how the empirical test error of the ensemble converges to the one of a single trained model when the number of parameters goes to infinity, leading to deterioration of the performance. In other words, the bigger the model, the harder it is to maintain diversity in the ensemble and avoid collapse to the same solution. This is intuitively due to the fact that bigger models are less sensitive to the initialization. Namely, in order for them to get stuck in a local minimum, they must have second derivatives that are positive simultaneously in all directions. As the number of hidden units gets larger, this becomes less likely.

2.1 Repulsive force in weight space

To overcome the aforementioned limitations of standard deep ensembles, we introduce, inspired by SVGD [43], a deep ensemble with members that interact with each other through a repulsive component. Using a kernel function to model this interaction, the single models repel each other based on their position in the weight space, so that two members can never assume the same weights. Considering a stationary kernel $k(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ acting in the parameter space of the neural networks, a repulsive term can be parameterized through its gradient:

$$\phi(\mathbf{w}_i^t) = \nabla_{\mathbf{w}_i^t} \log p(\mathbf{w}_i^t|\mathcal{D}) - \mathcal{R}\left(\left\{\nabla_{\mathbf{w}_i^t} k(\mathbf{w}_i^t, \mathbf{w}_j^t)\right\}_{j=1}^n\right) . \tag{3}$$

To get an intuition for the behavior of this repulsive term and its gradients, we can consider the RBF kernel $k(\mathbf{w}, \mathbf{w}') = \exp\big(-\frac{1}{h}||\mathbf{w}_i - \mathbf{w}_j||^2\big)$ with lengthscale h and notice how its gradient

$$\nabla_{\mathbf{w}_i^t} k(\mathbf{w}_i^t, \mathbf{w}_j^t) = \frac{\sum_{j=1}^n \frac{2}{h}(\mathbf{w}_i^t - \mathbf{w}_j^t) k(\mathbf{w}_i^t, \mathbf{w}_j^t)}{\sum_{j=1}^n k(\mathbf{w}_i^t, \mathbf{w}_j^t)}$$

drives $\mathbf{w}_i$ away from its neighboring members $\mathbf{w}_j$, thus creating a repulsive effect.

2.2 Repulsive force in function space

To overcome the aforementioned overparameterization issue, the update in Eq. (3) can be formulated in function space instead of weight space. Let $\boldsymbol{f} : \mathbf{w} \mapsto f(\cdot\,; \mathbf{w})$ be the map that maps a configuration of weights $\mathbf{w} \in \mathbb{R}^d$ to the corresponding neural network regression function and denote $\boldsymbol{f}_i := f(\cdot\,; \mathbf{w}_i)$ the function with a certain configuration of weights $\mathbf{w}_i$. We can now consider n particles in function space $\{\boldsymbol{f}_i\}_{i=1}^n$ with $\boldsymbol{f} \in \mathcal{F}$ and model their interaction with a general positive definite kernel $k(\cdot, \cdot)$. We also consider the implicit functional likelihood $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{f})$, determined by the measure $p(\boldsymbol{y}|\boldsymbol{x}, \mathbf{w})$ in the weight space, as well as the functional prior $p(\boldsymbol{f})$, which can either be defined separately (e.g., using a GP) or modeled as a push-forward measure of the weight-space prior $p(\mathbf{w})$. Together, they determine the posterior in function space $p(\boldsymbol{f}|\mathcal{D})$. The functional evolution of a particle can then be written as:

$$\begin{aligned} \boldsymbol{f}_i^{t+1} &\leftarrow \boldsymbol{f}_i^t + \epsilon_t \phi(\boldsymbol{f}_i^t) \\ \text{with} \qquad \phi(\boldsymbol{f}_i^t) &= \nabla_{\boldsymbol{f}_i^t} \log p(\boldsymbol{f}_i^t|\mathcal{D}) - \mathcal{R}\left(\left\{\nabla_{\boldsymbol{f}_i^t} k(\boldsymbol{f}_i^t, \boldsymbol{f}_j^t)\right\}_{j=1}^n\right) . \end{aligned} \tag{4}$$

However, computing the update in function space is neither tractable nor practical, which is why two additional considerations are needed. The first one regards the infinite dimensionality of function space, which we circumvent using a canonical projection into a subspace:

Definition 1 (Canonical projection)**.** *For any $A \subset \mathcal{X}$, let $\pi_A : \mathbb{R}^{\mathcal{X}} \to \mathbb{R}^A$ denote the canonical projection onto A, that is, $\pi_A(f) = \{f(a)\}_{a \in A}$.*

In other words, the kernel will not be evaluated directly in function space, but on the projection $k\big(\pi_B(f), \pi_B(f')\big)$, with B being a subset of the input space given by a batch of training datapoints. The second consideration is to project this update back into the parameter space and evolve a set of particles there, because ultimately we are interested in representing the functions by parametrized neural networks. For this purpose, we can use the Jacobian of the i-th particle as a projector:

$$\phi(\mathbf{w}_i^t) = \left(\frac{\partial \boldsymbol{f}_i^t}{\partial \mathbf{w}_i^t}\right)^\top \left[\nabla_{\boldsymbol{f}_i^t} \log p(\boldsymbol{f}_i^t|\mathcal{D}) - \mathcal{R}\left(\left\{\nabla_{\boldsymbol{f}_i^t} k(\pi_B(\boldsymbol{f}_i^t), \pi_B(\boldsymbol{f}_j^t))\right\}_{j=1}^n\right)\right] . \tag{5}$$

2.3 Comparison to Stein variational gradient descent

Note that our update is reminiscent of SVGD [43], which in parameter space can be written as:

$$\phi(\mathbf{w}_i^t) = \sum_{j=1}^n k(\mathbf{w}_i^t, \mathbf{w}_j^t) \nabla_{\mathbf{w}_i^t} \log p(\mathbf{w}_i^t|\mathcal{D}) + \sum_{j=1}^n \nabla_{\mathbf{w}_j^t} k(\mathbf{w}_j^t, \mathbf{w}_i^t) . \tag{6}$$

It is important to notice that here, the gradients are averaged across all the particles using the kernel matrix. Moving the inference from parameter to function space [60], leads to the update rule

$$\phi(\mathbf{w}_i^t) = \left(\frac{\partial \boldsymbol{f}_i^t}{\partial \mathbf{w}_i^t}\right)^\top \left(\frac{1}{n} \sum_{j=1}^n k(\boldsymbol{f}_i^t, \boldsymbol{f}_j^t) \nabla_{\boldsymbol{f}_j^t} \log p(\boldsymbol{f}_j^t|\mathcal{D}) + \nabla_{\boldsymbol{f}_j^t} k(\boldsymbol{f}_i^t, \boldsymbol{f}_j^t)\right) . \tag{7}$$

This way of averaging gradients using a kernel can be dangerous in high-dimensional settings, where kernel methods often suffer from the curse of dimensionality. Moreover, in Eq. (6), the posterior gradients of the particles are averaged using their similarity in weight space, which can be misleading in multi-modal posteriors. Worse yet, in Eq. (7), the gradients are averaged in function space and are then projected back using exclusively the i-th Jacobian, which can be harmful given that it is not guaranteed that distances between functions evaluated on a subset of their input space resemble their true distance. Our proposed method, on the other hand, does not employ any averaging of the posterior gradients and thus comes closest to the true particle gradients in deep ensembles.

3 Repulsive deep ensembles are Bayesian

So far, we represented the repulsive force as a general function of the gradients of a kernel. In this section, we show how to determine the explicit form of the repulsive term, such that the resulting update rule is equivalent to the discretization of the gradient flow dynamics of the KL divergence in Wasserstein space. We begin by introducing the concepts of particle approximation and gradient flow.

3.1 Particle approximation

A particle-based approximation of a target measure depends on a set of weighted samples $\{(x_i, w_i)\}_{i=1}^n$, for which an empirical measure can be defined as

$$\rho(x) = \sum_{i=1}^{n} w_i \, \delta(x - x_i) \,, \tag{8}$$

where $\delta(\cdot)$ is the Dirac delta function and the weights w_i satisfy $w_i \in [0, 1]$ and $\sum_{i=1}^n w_i = 1$. To approximate a target distribution $\pi(x)$ using the empirical measure, the particles and their weights need to be selected in a principled manner that minimizes some measure of distance between $\pi(x)$ and $\rho(x)$ (e.g., a set of N samples with weights $w_i = 1/N$ obtained using an MCMC method).

3.2 Gradient flow in parameter space

Given a smooth function $J : \mathbb{R}^d \to \mathbb{R}$ in Euclidean space, we can minimize it by creating a path that follows its negative gradient starting from some initial conditions x_0. The curve $x(t)$ with starting point x_0 described by that path is called *gradient flow*. The dynamics and evolution in time of a considered point in the space under this minimization problem can be described as the ODE[1]

$$\frac{dx}{dt} = -\nabla J(x) \,. \tag{9}$$

Analogously, we can consider a Riemannian manifold $\mathcal{M}$ equipped with the metric tensor $G(x)$ defined for all $x \in \mathcal{M}$. Here, $G(x) : \mathcal{T}_{\mathcal{W}} \times \mathcal{T}_{\mathcal{W}} \to \mathbb{R}$ defines a smoothly varying local inner product on the tangent space at each point of the manifold x. For manifolds over $\mathbb{R}^d$, the metric tensor is a positive definite matrix that defines local distances for infinitesimal displacements $d(x, x + dx) = \sqrt{dx^\top G(x) dx}$. Considering a functional $J : \mathcal{M} \to \mathbb{R}$, the evolution in Eq. (9) becomes

$$\frac{dx}{dt} = -G(x)^{-1} \nabla J(x) \,. \tag{10}$$

We see that the metric tensor of the manifold acts like a perturbation of the gradient. We can now extend this concept to the space of probability distributions (*Wasserstein gradient flow*) [2]. Let us consider the space of probability measures $\mathcal{P}_2(\mathcal{M})$, that is the set of probability measures with finite second moments defined on the manifold $\mathcal{M}$:

$$\mathcal{P}_2(\mathcal{M}) = \left\{ \rho : \mathcal{M} \to [0, \infty) \middle| \int_{\mathcal{M}} d\rho = 1, \ \int_{\mathcal{M}} |x|^2 \rho(x) dx < +\infty \right\} .$$

Taking $\Pi(\mu, \nu)$ as the set of joint probability measures with marginals μ, ν, we can define the Wasserstein metric on the space $\mathcal{P}_2(\mathcal{M})$ as:

$$W_2^2(\mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \int |x - y|^2 \, \mathrm{d}\pi(x, y) \,. \tag{11}$$

Considering the optimization problem of a functional $J : \mathcal{P}_2(\mathcal{M}) \to \mathbb{R}$, such as the KL divergence between the particle approximation in Eq. (8) and the target posterior $\pi(x)$,

$$\inf_{\rho \in \mathcal{P}_2(\mathcal{M})} D_{KL}(\rho, \pi) = \int_{\mathcal{M}} (\log \rho(x) - \log \pi(x)) \rho(x) \, \mathrm{d}x \ ,$$

the evolution in time of the measure ρ under the equivalent of the gradient, the Wasserstein gradient flow, is described by the *Liouville equation*:

$$\begin{aligned} \frac{\partial \rho(x)}{\partial t} &= \nabla \cdot \left(\rho(x) \nabla \frac{\delta}{\delta \rho} D_{KL}(\rho, \pi) \right) \\ &= \nabla \cdot \left(\rho(x) \nabla \big(\log \rho(x) - \log \pi(x) \big) \right) , \end{aligned} \tag{12}$$

[1] Together with the initial condition, this is know as the Cauchy problem.

where $\nabla \frac{\delta}{\delta \rho} D_{KL}(\rho, \pi) =: \nabla_{\mathcal{W}_2} D_{KL}(\rho, \pi)$ is the Wasserstein gradient and the operator $\frac{\delta}{\delta \rho} : \mathcal{P}_2(\mathcal{M}) \to \mathbb{R}$ represents the functional derivative or first variation (see Appendix C for more details). In the particular case of the KL functional, we can recover the Fokker-Planck equation,

$$\begin{aligned}\frac{\partial \rho(x)}{\partial t} &= \nabla \cdot \big(\rho(x) \nabla (\log \rho(x) - \log \pi(x))\big) \\ &= -\nabla \cdot \big(\rho(x) \nabla \log \pi(x)\big) + \nabla^2 \rho(x)\,,\end{aligned}$$

that admits as unique stationary distribution the posterior $\pi(x)$. The deterministic particle dynamics ODE [1] related to Eq. (12), namely mean-field Wasserstein dynamics, is then given by:

$$\frac{dx}{dt} = -\nabla \big(\log \rho(x) - \log \pi(x)\big)\,. \tag{13}$$

Considering a discretization of Eq. (13) for a particle system $\{x\}_{i=1}^n$ and small stepsize ϵ_t, we can rewrite Eq. (13) as:

$$x_i^{t+1} = x_i^t + \epsilon_t \big(\nabla \log \pi(x_i^t) - \nabla \log \rho(x_i^t)\big)\,. \tag{14}$$

Unfortunately, we do not have access to the analytical form of the gradient $\nabla \log \rho$, so an approximation is needed. At this point, it is crucial to observe the similarity between the discretization of the Wasserstein gradient flow in Eq. (14) and the repulsive update in Eq. (3) to notice how, if the kernelized repulsion is an approximation of the gradient of the empirical particle measure, the update rule minimizes the KL divergence between the particle measure and the target posterior. Different sample-based approximations of the gradient that use a kernel function have been recently studied. The simplest one is given by the kernel density estimation (KDE) $\tilde{\rho}_t(x) = \sum_{i=1}^n k(x, x_t^i)$, where $k(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and the gradient of its log density is given by [56]:

$$\nabla \log \rho(x_i^t) \approx \frac{\sum_{j=1}^n \nabla_{x_i^t} k(x_i^t, x_j^t)}{\sum_{j=1}^n k(x_i^t, x_j^t)}\,. \tag{15}$$

Using this approximation in Eq. (14) we obtain:

$$x_i^{t+1} = x_i^t + \epsilon_t \left(\nabla \log \pi(x_i^t) - \frac{\sum_{j=1}^n \nabla_{x_i^t} k(x_i^t, x_j^t)}{\sum_{j=1}^n k(x_i^t, x_j^t)} \right), \tag{16}$$

where, if we substitute the posterior for π, we obtain an expression for the repulsive force in Eq. (3). This shows that if the repulsive term is the normalized sum of the gradients $\mathcal{R}(x_i^t) = \beta(x_i^t) \sum_{j=1}^n \nabla_{x_i^t} k(x_i^t, x_j^t)$, we do not only encourage diversity of the ensemble members and thus avoid collapse, but surprisingly—in the asymptotic limit of $n \to \infty$, where the KDE approximation is exact—also converge to the true Bayesian posterior!

Nevertheless, approximating the gradient of the empirical measure with the KDE can lead to suboptimal performance, as already studied by Li and Turner [39]. They instead introduced a new *Stein gradient estimator* (SGE) that offers better performance, while maintaining the same computational cost. Even more recently, Shi et al. [55] introduced a spectral method for gradient estimation (SSGE), that also allows for a simple estimation on out-of-sample points. These two estimators can be used in Eq. (14), leading to the following update rules. The one using the Stein estimator, that we will call SGE-WGD, is:

$$x_i^{t+1} = x_i^t + \epsilon_t \left(\nabla \log \pi(x_i^t) + \sum_{j=1}^n (K + \eta \mathbb{I})_{ij}^{-1} \sum_{k=1}^n \nabla_{x_k^t} k(x_k^t, x_j^t) \right), \tag{17}$$

where K is the kernel Gram matrix, η a small constant and $\mathbb{I}$ the identity matrix. We can notice the important difference between KDE and SGE that the former one is only considering the interaction of the i-th particle being updated with all the others, while the latter is simultaneously considering also the interactions between the remaining particles. The spectral method, that we will call SSGE-WGD, leads to the following update rule:

$$x_{t+1}^i = x_t^i + \epsilon_t \left(\nabla \log \pi(x_t^i) + \sum_{j=1}^J \frac{1}{\lambda_j^2} \sum_{m=1}^n \sum_{k=1}^n u_{jk} \nabla_{x_m} k(x_m^t, x_k^t) \cdot \sum_{l=1}^n u_{jl} k(x_i^t, x_l^t) \right) \tag{18}$$

where λ_j is the j-th eigenvalue of the kernel matrix and u_{jk} is the k-th component of the j-th eigenvector.

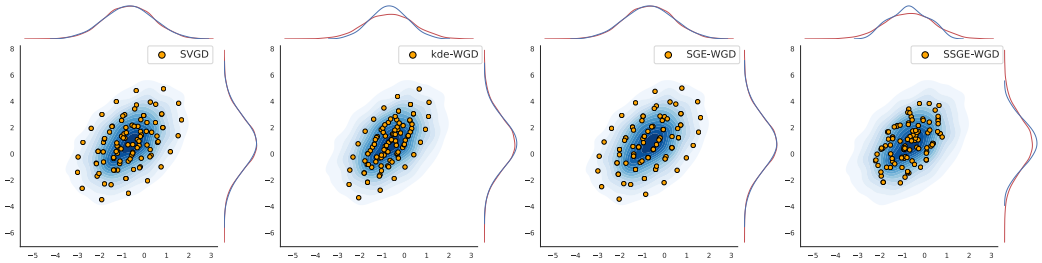

Figure 2: **Single Gaussian.** We show samples from SVGD, KDE-WGD, SGE-WGD, and SSGE-WGD (from left to right). The SGE-WGD and SVGD fit the target almost perfectly.

3.3 Gradient flow in function space

To theoretically justify the update rule introduced in function space in Eq. (5), we can rewrite the Liouville equation for the gradient flow in Eq. (12) in function space as

$$\begin{aligned} \frac{\partial \rho(\boldsymbol{f})}{\partial t} &= \nabla \cdot \left(\rho(\boldsymbol{f}) \nabla \frac{\delta}{\delta \rho} D_{KL}(\rho, \pi) \right) \\ &= \nabla \cdot \left(\rho(\boldsymbol{f}) \nabla \big(\log \rho(\boldsymbol{f}) - \log \pi(\boldsymbol{f}) \big) \right). \end{aligned} \tag{19}$$

Following this update, the mean field functional dynamics are

$$\frac{d\boldsymbol{f}}{dt} = -\nabla \big(\log \rho(\boldsymbol{f}) - \log \pi(\boldsymbol{f}) \big). \tag{20}$$

Using the same KDE approximation as above, we can obtain a discretized evolution in function space and with it an explicit form for the repulsive force in Eq. (4) as

$$\boldsymbol{f}_{t+1}^i = \boldsymbol{f}_t^i + \epsilon_t \left(\nabla_{\boldsymbol{f}} \log \pi(\boldsymbol{f}_t^i) - \frac{\sum_{j=1}^n \nabla_{\boldsymbol{f}_i^t} k(\boldsymbol{f}_i^t, \boldsymbol{f}_j^t)}{\sum_{j=1}^n k(\boldsymbol{f}_i^t, \boldsymbol{f}_j^t)} \right). \tag{21}$$

The update rules using the SGE and SSGE approximations follow as for the parametric case.

4 Experiments

In this section, we compare the different proposed WGD methods with deep ensembles and SVGD on synthetic sampling, regression, and classification tasks and real-world image classification tasks. We also quantitatively assess the uncertainty estimation of the methods in terms of calibration and OOD detection. In our experiments, we report the test accuracy, negative log-likelihood (NLL), and the expected calibration error (ECE) [49]. To assert the robustness on out-of-distribution (OOD) data, we report the ratio between predictive entropy on OOD and test data points (H_o/H_t), and the OOD detection area under the ROC curve AUROC(H) [40]. Moreover, to assess the diversity of the ensemble generated by the different methods in function space, we measure the functional diversity using the model disagreement (MD) (details in Appendix B). In particular, we report the ratio between the average model disagreement on the OOD and test data points (MD_o/MD_t) and additionally the AUROC(MD) computed using this measure instead of the entropy.

Sampling from synthetic distributions As a sanity check, we first assessed the ability of our different approximations for Wasserstein gradient descent (using KDE, SGE, and SSGE) to sample from a two-dimensional univariate Gaussian distribution (Figure 2). We see that our SGE-WGD and the SVGD fit the target almost perfectly. We also tested the different methods in a more complex two-dimensional Funnel distribution [50] and present the results in Figure E.1 in the Appendix. There, SGE-WGD and SVGD also perform best.

BNN 1D regression We then assessed the different methods in fitting a BNN posterior on a synthetically generated one-dimensional regression task. The results are reported in Figure 1, consisting of the mean prediction and $\pm 1, 2, 3$ standard deviations of the predictive distribution. We

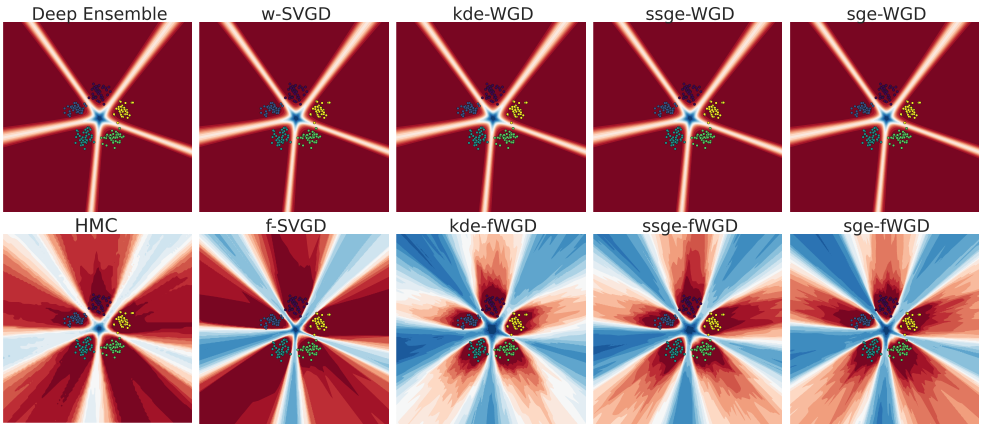

Figure 3: **BNN 2D classification.** Again, the function-space methods capture the epistemic uncertainty better than the weight-space ones, thus approaching the gold-standard HMC posterior.

can see that all methods performing inference in the weight space (DE, w-SVGD, WGD) are unable to capture the epistemic uncertainty between the two clusters of training data points. Conversely, the functional methods (f-SVGD, fWGD) are perfectly able to infer the diversity of the hypotheses in this region due to the lack of training evidence. They thereby achieve a predictive posterior that very closely resembles the one obtained with the gold-standard HMC sampling.

BNN 2D classification Next, we investigated the predictive performance and quality of uncertainty estimation of the methods in a two-dimensional synthetic classification setting. The results are displayed in Figure 3. We can clearly observe that the weight-space methods are overconfident and do not capture the uncertainty well. Moreover, all the functions seems to collapse to the optimal classifier. These methods thus only account for uncertainty close to the decision boundaries and the origin region, for which the uncertainty is purely aleatoric. In this setting, f-SVGD suffers for a similar issues as the weight space methods, being overconfident away from the training data. Conversely, our fWGD methods are confident (low entropy) around the data but not out-of-distribution, thus representing the epistemic uncertainty better. This suggests that the functional diversity captured by this method naturally leads to a *distance-aware* uncertainty estimation, a property that translates into confident predictions only in the proximity of the training data, allowing for a principled OOD detection.

FashionMNIST classification In this section, the methods are tested in an image classification setting using the FashionMNIST dataset [67] for training and the MNIST dataset [36] as an out-of-distribution (OOD) task. The results are reported in Table 1 (top). We can see that all our methods improve upon standard deep ensembles and SVGD, highlighting the effectiveness of our proposed repulsion terms when training neural network ensembles. In particular, the sge-WGD offers the best accuracy, whereas the methods in function space all offer a better OOD detection. This is probably due to the fact that these methods achieve an higher entropy ratio and functional diversity measured via the model disagreement when compared to their weight space counterparts. Interestingly, they also reach the lowest NLL values. We can also notice how the model disagreement (MD) not only serves its purpose as a metric for the functional heterogeneity of the ensemble but also allows for a better OOD detection in comparison to the entropy. To the best of our knowledge, this insight has not been described before.

CIFAR classification In this experiment, we use a ResNet32 architecture [24] on CIFAR-10 [34] with the SVHN dataset [52] as OOD data. The results are reported in Table 1 (bottom). We can see that in this case the weight space methods achieve better performance in accuracy and OOD detection using the entropy than the ones in function space. Nevertheless, all our repulsive ensembles improve functional diversity, accuracy, and OOD detection when compared to standard SVGD whereas the standard deep ensemble achieves the best OOD detection using the entropy.

Table 1: **BNN image classification. AUROC(H)** is the AUROC computed using the entropy whereas **AUROC(MD)** is computed using the model disagreement. $\mathbf{H_o/H_t}$ is the ratio of the entropies on OOD and test points respectively and $\mathbf{MD_o/MD_t}$ is the ratio for model disagreement. We see that the best accuracy is achieved by our WGD methods, while our fWGD methods yield the best OOD detection and funtional diversity. All our proposed methods improve over standard deep ensembles in terms of accuracy and diversity, highlighting the effect of our repulsion.

		AUROC(H)	AUROC(MD)	Accuracy	$\mathrm{H_o/H_t}$	$\mathrm{MD_o/MD_t}$	ECE	NLL
FashionMNIST	**Deep ensemble [35]**	0.958±0.001	0.975±0.001	91.122±0.013	6.257±0.005	6.394±0.001	**0.012±0.001**	0.129±0.001
	SVGD [43]	0.960±0.001	0.973±0.001	91.134±0.024	6.315±0.019	6.395±0.018	0.014±0.001	0.127±0.001
	f-SVGD [60]	0.956±0.001	0.975±0.001	89.884±0.015	5.652±0.009	6.531±0.005	0.013±0.001	0.150±0.001
	kde-WGD (ours)	0.960±0.001	0.970±0.001	91.238±0.019	6.587±0.019	6.379±0.018	0.014±0.001	0.128±0.001
	sge-WGD (ours)	0.960±0.001	0.970±0.001	**91.312±0.016**	6.562±0.007	6.363±0.009	**0.012±0.001**	0.128±0.001
	ssge-WGD (ours)	0.968±0.001	0.979±0.001	91.198±0.024	6.522±0.009	6.610±0.012	**0.012±0.001**	0.130±0.001
	kde-fWGD (ours)	**0.971±0.001**	**0.980±0.001**	91.260±0.011	7.079±0.016	6.887±0.015	0.015±0.001	**0.125±0.001**
	sge-fWGD (ours)	0.969±0.001	0.978±0.001	91.192±0.013	7.076±0.004	6.900±0.005	0.015±0.001	**0.125±0.001**
	ssge-fWGD (ours)	**0.971±0.001**	**0.980±0.001**	91.240±0.022	**7.129±0.006**	**6.951±0.005**	0.016±0.001	**0.124±0.001**
CIFAR10	**Deep ensemble [35]**	**0.843±0.004**	0.736±0.005	85.552±0.076	**2.244±0.006**	1.667±0.008	0.049±0.001	0.277±0.001
	SVGD [43]	0.825±0.001	0.710±0.002	85.142±0.017	2.106±0.003	1.567±0.004	0.052±0.001	0.287±0.001
	fSVGD [60]	0.783±0.001	0.712±0.001	84.510±0.031	1.968±0.004	1.624±0.003	0.049±0.001	0.292±0.001
	kde-WGD (ours)	0.838±0.001	0.735±0.004	**85.904±0.030**	2.205±0.003	1.661±0.008	0.053±0.001	**0.276±0.001**
	sge-WGD (ours)	0.837±0.003	0.725±0.004	85.792±0.035	2.214±0.010	1.634±0.004	0.051±0.001	**0.275±0.001**
	ssge-WGD (ours)	0.832±0.003	0.731±0.005	85.638±0.038	2.182±0.015	1.655±0.001	0.049±0.001	**0.276±0.001**
	kde-fWGD (ours)	0.791±0.002	**0.758±0.002**	84.888±0.030	1.970±0.004	**1.749±0.005**	**0.044±0.001**	0.282±0.001
	sge-fWGD (ours)	0.795±0.001	0.754±0.002	84.766±0.060	1.984±0.003	1.729±0.002	0.047±0.001	0.288±0.001
	ssge-fWGD (ours)	0.792±0.002	0.752±0.002	84.762±0.034	1.970±0.006	1.723±0.005	0.046±0.001	0.286±0.001

5 Related Work

The theoretical and empirical properties of SVGD have been well studied [33, 41, 12] and it can also be seen as a Wasserstein gradient flow of the KL divergence in the Stein geometry [13, 42]. Interestingly, a gradient flow interpretation is also possible for (stochastic gradient) MCMC-type algorithms [41], which can be unified under a general particle inference framework [9]. Moreover, our Wasserstein gradient descent using the SGE approximation can also be derived using an alternative formulation as a gradient flow with smoothed test functions [41]. A projected version of WGD has been studied in [59], which could also be readily applied in our framework. Besides particle methods, Bayesian neural networks MacKay [45], Neal [50] have gained popularity recently [63, 18, 16, 30], using modern MCMC [50, 63, 18, 20, 17] and variational inference techniques [4, 58, 14, 28]. Ensemble methods, on the other hand, have also been extensively studied [35, 15, 66, 19] and many variants have been proposed [64, 23, 26, 69, 62]. Moreover, providing Bayesian interpretations for deep ensembles has been previously attempted through the lenses of stationary SGD distributions [47, 7], ensembles of linear models [48], additional random functions [11, 23], approximate inference [66], and marginal likelihood lower bounds [44]. Furthermore, variational inference in function space has recently gained attention [57] and the limitations of the KL divergence have been studied in [6].

6 Conclusion

We have presented a simple and principled way to improve upon standard deep ensemble methods. To this end, we have shown that the introduction of a kernelized repulsion between members of the ensemble not only improves the accuracy of the predictions but—even more importantly—can be seen as Wasserstein gradient descent on the KL divergence, thus transforming the MAP inference of deep ensembles into proper Bayesian inference. Moreover, we have shown that incorporating functional repulsion between ensemble members can improve the quality of the estimated uncertainties on simple synthetic examples and OOD detection on real-world data and can approach the true Bayesian posterior more closely.

In future work, it will be interesting to study the impact of the Jacobian in the fWGD update and its implications on the Liouville equation in more detail, also compared to other neural network Jacobian methods, such as neural tangent kernels [31] and generalized Gauss-Newton approximations [29]. Moreover, it would be interesting to derive explicit convergence bounds for our proposed method and compare them to the existing bounds for SVGD [33].

References

[1] Luigi Ambrosio and Gianluca Crippa. Continuity equations and ode flows with non-smooth velocity. *Proceedings of the Royal Society of Edinburgh: Section A Mathematics*, 144(6): 1191–1244, 2014.

[2] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.

[3] Vijay Badrinarayanan, Bamdev Mishra, and Roberto Cipolla. Symmetry-invariant optimization in deep networks. *arXiv preprint arXiv:1511.01754*, 2015.

[4] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

[5] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[6] David R Burt, Sebastian W Ober, Adrià Garriga-Alonso, and Mark van der Wilk. Understanding variational inference in function-space. *arXiv preprint arXiv:2011.09421*, 2020.

[7] Pratik Chaudhari and Stefano Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–10. IEEE, 2018.

[8] An Mei Chen, Haw-minn Lu, and Robert Hecht-Nielsen. On the geometry of feedforward neural network error surfaces. *Neural computation*, 5(6):910–927, 1993.

[9] Changyou Chen, Ruiyi Zhang, Wenlin Wang, Bai Li, and Liqun Chen. A unified particle-optimization framework for scalable bayesian sampling. *arXiv preprint arXiv:1805.11659*, 2018.

[10] Jiefeng Chen, Xi Wu, Yingyu Liang, Somesh Jha, et al. Robust out-of-distribution detection in neural networks. *arXiv preprint arXiv:2003.09711*, 2020.

[11] Kamil Ciosek, Vincent Fortuin, Ryota Tomioka, Katja Hofmann, and Richard Turner. Conservative uncertainty estimation by fitting prior networks. In *International Conference on Learning Representations*, 2019.

[12] Francesco D'Angelo and Vincent Fortuin. Annealed stein variational gradient descent. *arXiv preprint arXiv:2101.09815*, 2021.

[13] A Duncan, Nikolas Nuesken, and Lukasz Szpruch. On the geometry of stein variational gradient descent. *arXiv preprint arXiv:1912.00894*, 2019.

[14] Michael W Dusenberry, Ghassen Jerfel, Yeming Wen, Yi-an Ma, Jasper Snoek, Katherine Heller, Balaji Lakshminarayanan, and Dustin Tran. Efficient and scalable bayesian neural nets with rank-1 factors. *arXiv preprint arXiv:2005.07186*, 2020.

[15] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

[16] Vincent Fortuin. Priors in bayesian deep learning: A review. *arXiv preprint arXiv:2105.06868*, 2021.

[17] Vincent Fortuin, Adrià Garriga-Alonso, Mark van der Wilk, and Laurence Aitchison. Bnnpriors: A library for bayesian neural network inference with different prior distributions. *Software Impacts*, page 100079, 2021.

[18] Vincent Fortuin, Adrià Garriga-Alonso, Florian Wenzel, Gunnar Rätsch, Richard Turner, Mark van der Wilk, and Laurence Aitchison. Bayesian neural network priors revisited. *arXiv preprint arXiv:2102.06571*, 2021.

[19] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *arXiv preprint arXiv:1802.10026*, 2018.

[20] Adrià Garriga-Alonso and Vincent Fortuin. Exact langevin dynamics with stochastic gradients. *arXiv preprint arXiv:2102.01691*, 2021.

[21] Mario Geiger, Arthur Jacot, Stefano Spigler, Franck Gabriel, Levent Sagun, Stéphane d'Ascoli, Giulio Biroli, Clément Hongler, and Matthieu Wyart. Scaling description of generalization with number of parameters in deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(2):023401, 2020.

[22] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(10):993–1001, October 1990. ISSN 0162-8828. doi: 10.1109/34.58871. URL https://doi.org/10.1109/34.58871.

[23] Bobby He, Balaji Lakshminarayanan, and Yee Whye Teh. Bayesian deep ensembles via the neural tangent kernel. *arXiv preprint arXiv:2007.05864*, 2020.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[25] Robert Hecht-Nielsen. On the algebraic structure of feedforward network weight spaces. In *Advanced Neural Computers*, pages 129–135. Elsevier, 1990.

[26] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017.

[27] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. Snapshot ensembles: Train 1, get M for free. *CoRR*, abs/1704.00109, 2017. URL http://arxiv.org/abs/1704.00109.

[28] Alexander Immer, Matthias Bauer, Vincent Fortuin, Gunnar Rätsch, and Mohammad Emtiyaz Khan. Scalable marginal likelihood estimation for model selection in deep learning. *arXiv preprint arXiv:2104.04975*, 2021.

[29] Alexander Immer, Maciej Korzepa, and Matthias Bauer. Improving predictions of bayesian neural nets via local linearization. In *International Conference on Artificial Intelligence and Statistics*, pages 703–711. PMLR, 2021.

[30] Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Wilson. What are bayesian neural network posteriors really like? *arXiv preprint arXiv:2104.14421*, 2021.

[31] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: convergence and generalization in neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 8580–8589, 2018.

[32] Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998.

[33] Anna Korba, Adil Salim, Michael Arbel, Giulia Luise, and Arthur Gretton. A non-asymptotic analysis for stein variational gradient descent. *Advances in Neural Information Processing Systems*, 33, 2020.

[34] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

[35] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413, 2017.

[36] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[37] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

[38] E. Levin, N. Tishby, and S. A. Solla. A statistical approach to learning and generalization in layered neural networks. *Proc. of the IEEE – Special issue on Neural Networks*, 1990.

[39] Yingzhen Li and Richard E Turner. Gradient estimators for implicit models. *arXiv preprint arXiv:1705.07107*, 2017.

[40] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[41] Chang Liu, Jingwei Zhuo, Pengyu Cheng, Ruiyi Zhang, and Jun Zhu. Understanding and accelerating particle-based variational inference. In *International Conference on Machine Learning*, pages 4082–4092. PMLR, 2019.

[42] Qiang Liu. Stein variational gradient descent as gradient flow. In *Advances in neural information processing systems*, pages 3115–3123, 2017.

[43] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in neural information processing systems*, pages 2378–2386, 2016.

[44] Clare Lyle, Lisa Schut, Robin Ru, Yarin Gal, and Mark van der Wilk. A bayesian perspective on training speed and model selection. *Advances in Neural Information Processing Systems*, 33, 2020.

[45] David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

[46] Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble distribution distillation. *arXiv preprint arXiv:1905.00076*, 2019.

[47] Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *arXiv preprint arXiv:1704.04289*, 2017.

[48] Alexander G de G Matthews, Jiri Hron, Richard E Turner, and Zoubin Ghahramani. Sample-then-optimize posterior sampling for bayesian linear models. In *NeurIPS Workshop on Advances in Approximate Bayesian Inference*, 2017.

[49] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[50] Radford M Neal. Bayesian learning for neural networks. 1995.

[51] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

[52] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[53] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pages 13991–14002, 2019.

[54] Alexandre Rame and Matthieu Cord. Dice: Diversity in deep ensembles via conditional redundancy adversarial estimation. *arXiv preprint arXiv:2101.05544*, 2021.

[55] Jiaxin Shi, Shengyang Sun, and Jun Zhu. A spectral approach to gradient estimation for implicit distributions. In *International Conference on Machine Learning*, pages 4644–4653. PMLR, 2018.

[56] Radhey S Singh. Improvement on some known nonparametric uniformly consistent estimators of derivatives of a density. *The Annals of Statistics*, pages 394–399, 1977.

[57] Shengyang Sun, Guodong Zhang, Jiaxin Shi, and Roger Grosse. Functional variational bayesian neural networks. *arXiv preprint arXiv:1903.05779*, 2019.

[58] Jakub Swiatkowski, Kevin Roth, Bastiaan S Veeling, Linh Tran, Joshua V Dillon, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. The k-tied normal distribution: A compact parameterization of gaussian mean field posteriors in bayesian neural networks. *arXiv preprint arXiv:2002.02655*, 2020.

[59] Yifei Wang, Peng Chen, and Wuchen Li. Projected wasserstein gradient descent for high-dimensional bayesian inference. *arXiv preprint arXiv:2102.06350*, 2021.

[60] Ziyu Wang, Tongzheng Ren, Jun Zhu, and Bo Zhang. Function space particle optimization for bayesian neural networks. *arXiv preprint arXiv:1902.09754*, 2019.

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

[62] Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. *arXiv preprint arXiv:2002.06715*, 2020.

[63] Florian Wenzel, Kevin Roth, Bastiaan S Veeling, Jakub Światkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? *arXiv preprint arXiv:2002.02405*, 2020.

[64] Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. Hyperparameter ensembles for robustness and uncertainty quantification. In *Advances in Neural Information Processing Systems*, 2020.

[65] Christopher KI Williams. Computation with infinite neural networks. *Neural Computation*, 10 (5):1203–1216, 1998.

[66] Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*, 2020.

[67] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[68] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[69] Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient mcmc for bayesian deep learning. *arXiv preprint arXiv:1902.03932*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] In the supplemental material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In the supplemental material
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] All tables report the standard errors over multiple runs
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] see Sec. F in the supplemental material
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cited all the datasets
   (b) Did you mention the license of the assets? [No] They are standard benchmark datasets
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We release code
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]