# Repulsive Deep Ensembles are Bayesian

**Polina Barabanshchikova**
Institut Polytechnique de Paris

**David Faget Caño**
ENS Paris-Saclay

**Gabriel Singer**
ENS Paris-Saclay

## Abstract

Deep ensembles, known for their straightforward design and effectiveness, face challenges in preserving functional diversity among independently gradient-trained members. This issue can lead to performance saturation with additional members, affecting not just prediction quality but also uncertainty estimates and out-of-distribution data detection. The paper [1] under study propose addressing this by introducing a kernelized repulsive term to the ensemble update rule, preventing members from converging to identical functions. This modification not only ensures diversity but also shifts maximum a posteriori inference towards genuine Bayesian inference by aligning training dynamics with a Wasserstein gradient flow of the KL divergence with the true posterior. Authors explore repulsive terms in both weight and function spaces, conducting empirical comparisons with standard ensembles and Bayesian methods across various synthetic and real-world tasks.

## 1 Introduction

The paper [1] under review outlines the context and objectives of the study in the field of approximate Bayesian inference and deep learning. It acknowledges recent theoretical advancements in sampling algorithms for Bayesian inference, notably the reinterpretation of Markov Chain Monte Carlo methods as gradient flows of the KL divergence over Wasserstein space. In line with this research, a novel particle-optimization variational inference method, called SVGD [7], uses a kernel in parameter space to introduce a repulsive force, guiding samples towards high-density regions of the target distribution without collapsing to a single point. This concept of repulsion is central to the paper's investigation.

The paper focuses on deep ensembles [6], which are ensembles of neural networks successful in predictive performance and uncertainty estimation. However, existing deep ensembles lack guarantees for diversity between individual models and do not provably converge to the true Bayesian posterior. To address these limitations, the paper introduces a repulsive term between ensemble members, inspired by SVGD. This addition not only ensures diversity among members, but also allows to reformulate the method as a gradient flow of the KL divergence in the Wasserstein space, providing convergence guarantees to the true Bayesian posterior.

Another problem addressed is the non-identifiability in Bayesian Neural Networks (BNNs) due to overparametrization. Different weight configurations might map to the same function, creating a false sense of diversity. This leads to redundant posterior distributions and poor uncertainty estimation. To counter this, the authors propose an update rule for repulsive ensembles that approximates the gradient flow of the KL divergence in function space.

We explain the context of the paper in Section 2. In Section 3, we expose the strong and weak points of this work, and we present our experimental results in Section 4. We trained and evaluated deep repulsive ensembles on the MNIST dataset with NotMNIST test split as out-of-distribution (OOD) data, and assessed model calibration on RotatedMNIST. Expanding the original pipeline, we integrated adversarial training and added an option to use Laplace kernel instead of RBF. Furthermore, we investigated the impact of ensemble size on metrics and introduced novel visualizations. [1]

[1] https://github.com/pollinab/repulsive_ensembles

## 2 Explanation of the Paper

In supervised deep learning, we typically use a likelihood function $p(y|f(x;w))$, parameterized by a neural network $f(x;w)$ and training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Bayesian Neural Networks (BNNs) focus on the posterior distribution of network weights $p(w|\mathcal{D})$ proportional to $\prod_{i=1}^n p(y_i|f(x_i;w))p(w)$, where $p(w)$ is the prior distribution over weights. BNNs integrate over the entire posterior for predictions on a test point $x_*$:

$$p(y_*|x_*, \mathcal{D}) = \int p(y_*|f(x_*;w))p(w|\mathcal{D})dw. \tag{1}$$

However, approximating or sampling from the BNN posterior is challenging. As an alternative, a deep ensemble approach proposes to train multiple neural networks to approximate the posterior mode (MAP). Given the non-convex nature of MAP optimization, these networks converge to potentially diverse solutions across the posterior distribution landscape. Ensembles then combine predictions of different members to create a predictive distribution by computing the Bayesian Model Average as in (1). Evidently, effectiveness of ensembles relies on the diversity of their members. To ensure diversity, the authors introduce a repulsive deep ensemble inspired by SVGD. Its members interact via a repulsive component preventing them from assuming identical weights. This repulsive term $R$ is modeled by a gradient of a stationary kernel function $k(\cdot,\cdot)$, leading to the following update:

$$w_i^{t+1} \leftarrow w_i^t + \epsilon_t \phi(w_i^t) \quad \text{with} \quad \phi(w_i^t) = \nabla_{w_i^t} \log p(w_i^t|\mathcal{D}) - R\left(\sum_{j=1}^n \nabla_{w_i^t} k(w_i^t, w_j^t)\right). \tag{2}$$

To circumvent overparameterization, they formulate a similar update rule in function space, mapping weights $w$ to neural network functions $f(\cdot;w)$. The interaction among functions are modeled by a general positive definite kernel $k(\cdot,\cdot)$, with the functional evolution described by:

$$f_i^{t+1} \leftarrow f_i^t + \epsilon_t \phi(f_i^t) \quad \text{with} \quad \phi(f_i^t) = \nabla_{f_i^t} \log p(f_i^t|\mathcal{D}) - R\left(\sum_{j=1}^n \nabla_{f_i^t} k(f_i^t, f_j^t)\right). \tag{3}$$

As a direct computation of the repulsive term is impractical, the authors project functions on a batch of training points before evaluating the kernel. Finally, using the Jacobian of the $i$-th particle, they project the update back into the parameter space.

To show that repulsive deep ensembles converge to the Bayes posterior, the authors connect the update rule (2) with a gradient flow dynamics minimizing the KL divergence in Wasserstein space. For a set of weighted samples $\{(x_i, \alpha_i)\}_{i=1}^n$, they define a particle approximation of a target measure as an empirical distribution: $\rho(x) = \sum_{i=1}^n \alpha_i \delta(x - x_i)$. The evolution of the measure $\rho$ under Wasserstein gradient flow of the KL divergence between $\rho$ and the target posterior $\pi$ is described by a deterministic particle dynamics ODE. Using a kernel density estimator (KDE), the authors approximate the gradient of $\log \rho$ and obtain a discretization of this ODE:

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

Similarly, the update rule introduced in function space (3) is theoretically justified by reinterpreting the Liouville equation for gradient flow in function space. Thus, for a particular choice of the repulsive term, deep ensembles provably converge to the true Bayes posterior. Finally, the authors evaluate WGD methods against deep ensembles and SVGD across synthetic and real-world tasks, focusing on sampling, regression, classification, and image classification problems. In particular:

1. BNN Regression: In one-dimensional regression, functional methods outperformed weight-space methods in capturing epistemic uncertainty.
2. FashionMNIST Classification: On FashionMNIST, with MNIST as OOD data, proposed methods improved upon deep ensembles and SVGD in terms of accuracy and OOD detection.
3. CIFAR Classification: Using ResNet32 on CIFAR-10, with SVHN as OOD data, weight-space methods performed better in accuracy and OOD detection than functional space methods.

## 3 Strong and Weak Points of the Paper

### 3.1 Theoretical contribution

The main theoretical contribution of the paper lies in linking deep repulsive ensembles with the approximate Wasserstein gradient flow of the KL divergence. It bridges the gap between two previously successful approaches: deep ensembles and SVGD, incorporating the strengths of both. Specifically, repulsive update endow deep ensembles with Bayesian convergence properties and, unlike SVGD, which employs a kernel matrix to average gradients across all particles, this approach does not suffer from the curse of dimensionality.

A weakness of the paper is the absence of explicit convergence rates or bounds similar to those of SVGD presented in [5]. It remains unclear how many particles are needed to ensure a good approximation of the Bayes posterior and whether this number is feasible. Understanding the dependence of the convergence rate on the choice of kernel function and gradient estimator would offer valuable insights for practitioners in selecting the preferred method.

### 3.2 Practical potential

The authors support their theoretical insights with experimental evidence. They successfully tested their algorithm against deep ensemble and SVGD baselines on low-dimensional synthetic and high-dimensional image data, including FashionMNIST and CIFAR for classification and OOD detection tasks. This highlights the algorithm's applicability to real-world problems.

From a critical perspective, we observe that for CIFAR-10, the best reported accuracy of $85.9\%$ is significantly lower than current state-of-the-art ($> 99\%$). This relatively poor performance can be attributed to the absence of batch-normalization and data augmentation in the training pipeline. Since these practices cause deviation from the true Bayes posterior [9], they are usually omitted in BNN works. Notably, repulsive ensembles stand comparison with other Bayesian approaches without data augmentation, such as [8], achieving $86.7\%$ accuracy. However, we believe that it is important to demonstrate practical potential of the proposed approach and its compatibility with modern deep learning techniques. Additional experiments with data augmentation, as done in [8], would have undoubtedly strengthened the paper.

A strong point of the deep ensemble algorithm lies in its independence from the specific architecture of a single model. This property extends to repulsive ensembles, that are compatible with MLP, CNNs, or any other particles. However, it is crucial to note that when the repulsive force is computed in weight space, all particles should share the same architecture. Interestingly, this limitation doesn't apply to updates in function space, when the kernel is evaluated on the canonical projections of functions rather than their weights.

### 3.3 Computational overhead

Despite enhancing the performance of deep ensembles, the addition of the repulsive force comes with an extra computational cost of $\mathcal{O}(M^3 + M^2 d)$ for one iteration, with $M$ being a number of particles and $d$ their dimensionality. Hence, using $M \gg 1$ as the theoretical guaranties suggest might be impractical in most applications. Similarly, employing deep architectures with $d > 10^6$ may significantly slow down repulsive update in weight space.

Furthermore, for the theoretical guarantees to hold, all particles must be updated at the same time. This implies that before proceeding to the next iteration, each particle must receive an update. Therefore, parallel computing of the repulsive algorithm is complex, and sequential training of individual models is impossible. However, it's not mandatory for all particles to be present in memory simultaneously. For a repulsive update in function space, all weights can be saved on a hard disk and sequentially brought into and removed from the main memory for their updates. Nevertheless, this approach introduces additional computational time due to memory access overhead.

### 3.4 Omitted experimental details

From a critical standpoint, the experimental section of the paper lacks an analysis of the effect of the ensemble size on the predictive performance. Practical evidence demonstrating that the

metrics improve as a number of models increase would have supported the theoretical findings and strengthened the paper.

Additionally, while experimenting with our version of the repulsive ensemble, we noticed that the driving term ($\nabla \log p(w|\mathcal{D})$) in the gradient update is typically 6 orders of magnitude higher than the repulsive term. Running the authors' scripts with default parameters showed a similar behavior. This raises concerns about the extent to which the repulsive update affects the training of deep ensembles. We also noted that authors' implementation allows for multiplying the driving term by an annealing factor, increasing the influence of repulsion as training progresses. Unfortunately, we haven't found any details on this method in the paper.

### 3.5 Further directions

One of the paper's strength lies in its capacity to pave the way for further research. The authors encourage further investigations into the choice of kernel and the use of alternative priors in function space. Moreover, they suggest studying the impact of the Jacobian in the fWGD update in more detail, exploring possible connections with neural tangent kernels [4] and generalized Gauss-Newton approximations [3]. We think that exploring the possibility of training particles in a completely sequential manner also presents an interesting direction for future work. This involves modifying the regularization term to repel the particles from those that have already been trained to convergence.

## 4 Experiments

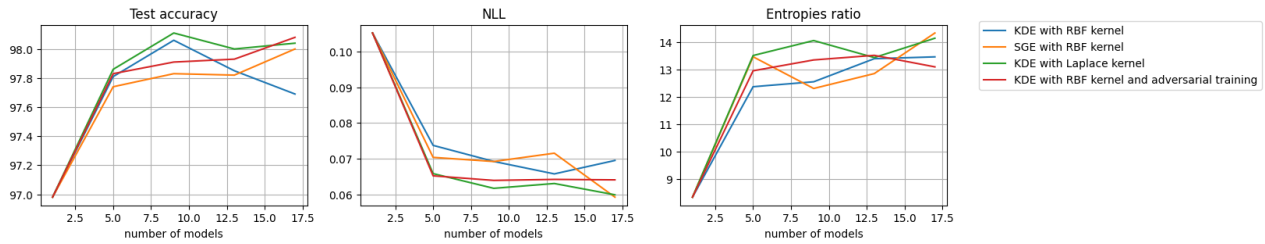



Figure 1: Evaluating predictive performance and uncertainty as a function of ensemble size.

In this section, we describe our experiments on real-world image classification and OOD detection tasks. The code and notebooks with additional figures are available in our GitHub repository: https://github.com/pollinab/repulsive_ensembles.

### 4.1 Implementation

Expanding upon the authors' repository, we developed our version of the training pipeline for deep ensembles with repulsive update in weight space (KDE, SGE, SSGE) and in function space (f-KDE, f-SGE, f-SSGE). In addition to the original RBF kernel, our implementation supports the Laplace kernel. Furthermore, motivated by the findings in [6], we added an option to use adversarial examples during training. Although adversarial training violates the likelihood principle, it has been demonstrated to improve predictive uncertainty and smooth the predictive distribution of deep ensembles. We employed fast gradient sign method introduced in [2] to generate adversarial examples. Lastly, we multiplied the driving term by a factor of $0.01$ to increase the importance of the repulsive component in the gradient update.

We trained an ensemble of MLP models on the MNIST dataset of hand-written digits and evaluated it on the test set with the NotMNIST dataset of letters as OOD data. Additionally, we assessed the calibration of our model on RotatedMNIST data containing randomly rotated and flipped digits.

### 4.2 Results

We examined how the methods' performance evolves with an increasing number of models. For each ensemble, we computed test accuracy, negative log-likelihood (NLL), expected calibration error

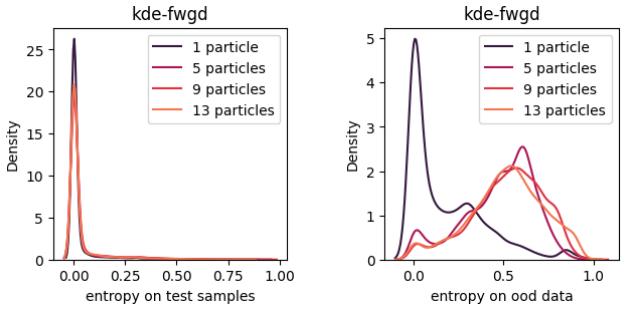
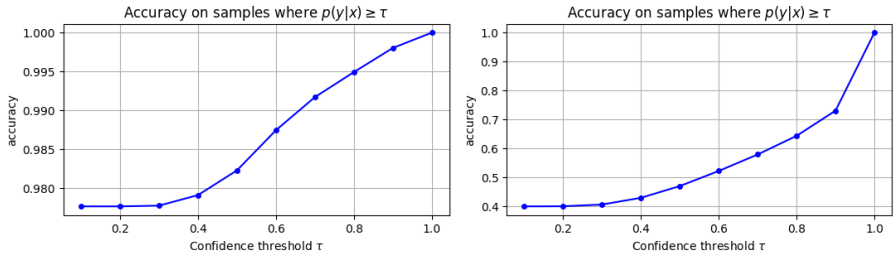



Figure 2: Histogram of the predictive entropy on test and OOD examples, as we vary ensemble size.




(a) MNIST test (b) RotatedMNIST

Figure 3: Accuracy vs Confidence curve.

(ECE), and the ratio between predictive entropy on OOD and test data points. The progression of these metrics is reported in Figure 1. While we observe an improvement of the ensemble approach over a single model, there is no clear trend in metrics as we increase the number of models from 5 to 17. The same holds for the ECE metric and the function-space methods. Interestingly, KDE with the Laplace kernel consistently outperformed KDE with the RBF kernel. Adversarial training enhances accuracy but does not seem to have a positive impact on predictive uncertainty.

Figure 2 illustrates the entropy of the predictive distribution of the repulsive ensemble on test examples and NotMNIST out-of-distribution data. Notably, a single model tends to provide overconfident predictions for unknown data, while an ensemble effectively captures uncertainty.

In Figure 3, we present the accuracy on the test samples $x$ with maximum predictive probability $\max_k p(y = k|x) \geq \tau$ plotted against a confidence threshold $\tau$. For a well-calibrated model, the accuracy should be at least as high as the corresponding confidence. This property holds for the test subset but not for the rotated data, possibly because certain digits can change classes under rotation.

Lastly, Figure 4 showcases examples of test and OOD data with low and high disagreement between the ensemble members. The disagreement is measured as a standard deviation between predictions and captures meaningful ambiguity: the ensemble agrees on images more typical to training data and disagrees on dissimilar examples. For examples sorted by entropy or confidence, please refer to our GitHub repository.

(a) MNIST test (b) RotatedMNIST (c) NotMNIST

Figure 4: Examples with low (top rows) and high (bottom rows) disagreement between the particles.

## References


[1] D'ANGELO, F., AND FORTUIN, V. Repulsive deep ensembles are bayesian, 2023.

[2] GOODFELLOW, I. J., SHLENS, J., AND SZEGEDY, C. Explaining and harnessing adversarial examples, 2015.

[3] IMMER, A., BAUER, M., FORTUIN, V., RÄTSCH, G., AND KHAN, M. E. Scalable marginal likelihood estimation for model selection in deep learning, 2021.

[4] JACOT, A., GABRIEL, F., AND HONGLER, C. Neural tangent kernel: Convergence and generalization in neural networks, 2020.

[5] KORBA, A., SALIM, A., ARBEL, M., LUISE, G., AND GRETTON, A. A non-asymptotic analysis for stein variational gradient descent, 2021.

[6] LAKSHMINARAYANAN, B., PRITZEL, A., AND BLUNDELL, C. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.

[7] LIU, Q., AND WANG, D. Stein variational gradient descent: A general purpose bayesian inference algorithm, 2019.

[8] OBER, S. W., AND AITCHISON, L. Global inducing point variational posteriors for bayesian neural networks and deep gaussian processes, 2021.

[9] WENZEL, F., ROTH, K., VEELING, B. S., ŚWIĄTKOWSKI, J., TRAN, L., MANDT, S., SNOEK, J., SALIMANS, T., JENATTON, R., AND NOWOZIN, S. How good is the bayes posterior in deep neural networks really?, 2020.