Function Space Diversity for Uncertainty Prediction via Repulsive Last-Layer Ensembles

Sophie Steger [1] **Christian Knoll** [2] **Bernhard Klein** [3] **Holger Fröning** [3] **Franz Pernkopf** [1]

Abstract

Bayesian inference in function space has gained attention due to its robustness against overparameterization in neural networks. However, approximating the infinite-dimensional function space introduces several new challenges. In this work, we discuss function space inference via particle optimization and present practical modifications that improve uncertainty estimation and, most importantly, make it applicable for large and pretrained networks. First, we demonstrate that the input samples, where particle predictions are enforced to be diverse, are detrimental to the model performance. While diversity on training data itself can lead to underfitting, the use of label-destroying data augmentation, or unlabeled out-of-distribution data can improve prediction diversity and uncertainty estimates. Furthermore, we take advantage of the function space formulation, which imposes no restrictions on network parameterization other than sufficient flexibility. Instead of using full deep ensembles to represent particles, we propose a single multi-headed network that introduces a minimal increase in parameters and computation. This allows seamless integration to pretrained networks, where this repulsive last-layer ensemble can be used for uncertainty aware fine-tuning at minimal additional cost.

1. Introduction

Particle-optimization variational inference (POVI) approximates the posterior distribution of Bayesian neural networks (BNNs) using a set of discrete and interacting particles (Liu & Wang, 2016; Liu, 2017; Liu et al., 2019). For deep ensembles (DEs), this entails modifying the optimization procedure by incorporating a kernelized *repulsion term* to enforce diverse ensemble members. Although the particles are distant in the parameter space, they might still correspond to similar prediction functions because of the overparamterized nature of neural networks (NNs). One way to avoid this issue and to achieve truly diverse particles is to perform inference directly in the space of functions, explicitly enforcing prediction diversity (Wang et al., 2019; D'Angelo & Fortuin, 2021). Despite its appeal, function space POVI has not been able to empirically outperform standard DEs regarding accuracy and uncertainty prediction (D'Angelo & Fortuin, 2021; Trinh et al., 2023; Yashima et al., 2022). In this work, we show that the lack of empirical performance is not because of the function space formulation itself, but rather because of approximation errors of the infinite-dimensional function space.

First, we highlight the importance of choosing appropriate input samples that lead to particles with diverse predictions; we refer to these as *repulsion samples*. It remains practically infeasible to achieve function space diversity over the whole input domain (particularly for high dimensional input data). Therefore, good repulsion samples must not only be *diverse* but also capture the most *relevant parts* of the input domain. The training data itself is generally not rich enough and, as such, insufficient for accurate uncertainty estimation (D'Angelo & Fortuin, 2021; Trinh et al., 2023). We show that the use of unlabeled out-of-distribution (OOD) data significantly improves uncertainty estimates without compromising domain accuracy. If OOD data is unavailable, label-destroying data augmentation[1] achieves similar quality. Enforcing diversity within those samples reduces the effect of spurious features and improves uncertainty calibration and OOD detection on unseen distributions.

Second, we address the computational limitations of training and storing DEs. For large models, training and maintaining multiple copies of the network may not be feasible; this is particularly problematic as the interactive repulsion term requires joint optimization of the whole ensemble. We exploit the fact that the function-space formulation of the inference problem does not impose any constraints on its parameteri-

[1]Graz University of Technology, Austria [2]Levata, Graz, Austria [3]University of Heidelberg, Germany. Correspondence to: Sophie Steger <sophie.steger@tugraz.at>.

Accepted by the Structured Probabilistic Inference & Generative Modeling workshop of ICML 2024, Vienna, Austria.

[1]Modification of input samples such that the original labels do not apply, e.g., shuffling of random image patches.

zation, besides requiring sufficient flexibility. Inspired by ensemble distillation methods (Tran et al., 2020), we propose a multi-headed network architecture, where each head represents a particle in function space. This allows utilizing a base network that learns a shared representation, and thus drastically reduces computational and memory costs. Prediction diversity is not introduced by random weight initialization but by the repulsion term in function space. This also allows for seamless integration with pre-trained networks, where the repulsive last-layer ensemble (RLL-E) is used for uncertainty-aware fine-tuning of the model.

We empirically evaluate our method for regression and classification tasks on synthetic and real-world datasets. We show that our network is able to (i) disentangle aleatoric and epistemic uncertainty for active learning (App. C.3), (ii) improve detection of both near and far OOD data (App. C.4), and (iii) provide calibrated uncertainty estimates under distribution shifts (App. C.5). Related work is summarized in App. D.

2. Background

We consider supervised learning tasks. Let $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N = (\mathbf{X}, \mathbf{Y})$ denote the training data set consisting of N i.i.d. data samples with inputs $\mathbf{x}_i \in \mathcal{X}$ and targets $\mathbf{y}_i \in \mathcal{Y}$. We define a likelihood model $p(\mathbf{y}|\mathbf{x}, \theta)$ with the mapping $f(\cdot;\theta) : \mathcal{X} \to \mathbb{R}^K$ parameterized by a NN.

2.1. Bayesian neural networks (BNNs)

BNNs treat the network parameters θ as random variables instead of point estimates. This entails defining a prior distribution of the parameters $p(\theta)$ to infer the posterior distribution of the parameters $p(\theta|\mathcal{D}) \propto p(\theta)p(\mathbf{Y}|\mathbf{X}, \theta)$. Predictions for a test data point $\mathbf{x}_{\text{test}}$ are obtained by marginalizing over all possible parameters $p(\mathbf{y}_{\text{test}}|\mathbf{x}_{\text{test}}, \mathcal{D}) = \int_\theta p(\mathbf{y}_{\text{test}}|\mathbf{x}_{\text{test}}, \theta)p(\theta|\mathcal{D})d\theta$.

2.2. Particle-optimization variational inference (POVI)

Variational inference approximates the posterior $p(\theta|\mathcal{D})$ by a simpler parametric distribution $q(\theta)$. POVI methods (Liu & Wang, 2016; Chen et al., 2018) aim to provide more flexibility by considering a non-parametric distribution, specified by a discrete set of particles $\{\theta^{(i)}\}_{i=1}^n$ according to $q(\theta) \approx \frac{1}{n}\sum_{i=1}^n \delta(\theta - \theta^{(i)})$, where $\delta(\cdot)$ is the Dirac function. The particles can then be optimized iteratively via

$$\theta_{l+1}^{(i)} \leftarrow \theta_l^{(i)} + \epsilon_l \mathrm{v}(\theta_l^{(i)}) \tag{1}$$

where ϵ_l is the step size at time step l. By viewing the particle optimization as a gradient flow in Wasserstein space, D'Angelo & Fortuin (2021) derive the following update rule that decomposes into an attraction and repulsion term

$$\mathrm{v}(\theta_l^{(i)}) = \underbrace{\nabla_{\theta_l^{(i)}} \log p(\theta_l^{(i)}|\mathcal{D})}_{\text{ATTRACTION}} - \underbrace{\frac{\sum_{j=1}^n \nabla_{\theta_l^{(i)}} k\left(\theta_l^{(i)}, \theta_l^{(j)}\right)}{\sum_{j=1}^n k\left(\theta_l^{(i)}, \theta_l^{(j)}\right)}}_{\text{REPULSION}} \tag{2}$$

where $k(\cdot,\cdot)$ denotes a kernel function. The attraction term drives particles into high-density regions of the posterior distribution, while the repulsion term induces diversity by preventing particles from collapsing into the same optima. The training procedure reduces to the standard maximum a posteriori (MAP) training in the one-particle limit, and it converges to the posterior distribution for $n \to \infty$ and a properly defined kernel (D'Angelo & Fortuin, 2021).

2.3. Diversity of prediction functions

The distance kernel needs to capture the variations in the predictions to enforce diverse particle *predictions* effectively. To achieve this objective, inference should be performed directly in the function space, with the kernel functions incorporating the predictions (Wang et al., 2019; D'Angelo & Fortuin, 2021). Then, the n particles represent functions $f^{(1)}(\mathcal{X}), \ldots, f^{(n)}(\mathcal{X})$ that are updated as

$$f_{l+1}^{(i)}(\mathcal{X}) \leftarrow f_l^{(i)}(\mathcal{X}) + \epsilon_l \mathrm{v}(f_l^{(i)}(\mathcal{X})). \tag{3}$$

However, to solve the problem we must rely on gradient based optimization procedures that in turn require a parameterized representation of the particles.

Function parameterization Each particle $f^{(i)}(\mathcal{X})$ is represented by a specific parameterization $f^{(i)}(\mathcal{X};\theta^{(i)})$. The parameterization $f^{(i)}(\mathcal{X};\theta^{(i)})$ must be sufficiently flexible to effectively approximate the underlying function space.

Repulsion samples Moreover, it remains prohibitive to evaluate $f^{(i)}(\mathcal{X};\theta^{(i)})$ across the entire input domain $\mathcal{X}$. Instead, prior work (Wang et al., 2019) adopted a mini-batch approximation, where the evaluation over the full set $\mathcal{X}$ is replaced with B *repulsion samples* drawn from an arbitrary distribution $\mathbf{x}_{rep} \sim \mu$ with support on $\mathcal{X}^B$. The variational distribution is shown to converge to the true posterior if the posterior is determined by almost all B-dimensional marginals $\{p(f(\mathbf{x})|\mathbf{X}, \mathbf{Y}) : \mathbf{x} \in \text{supp}(\mu)\}$ (Wang et al., 2019).

We summarize the importance of incorporating a repulsion term for a finite number of particles in App. A.

3. Improving function space approximations: Practical choices and implications

The function space formulation effectively circumvents issues regarding overparameterization and identifiability (Kirsch, 2024). Still, empirical results lack improvement over unregularized DEs, especially for large-scale image tasks (Wang et al., 2019; D'Angelo & Fortuin, 2021). In the following section, we show that these results are expected given previous choices for the approximations of the function space. Additionally, we highlight possibilities to perform function space inference with minimal computational overhead compared to single neural networks.

3.1. Choice of function parameterization

A key benefit of function space inference entails the theoretical justification to use any flexible network parameterization. Still, prior work has utilized the same DE structure, where each particle is parameterized by a separate neural network (Wang et al., 2019; D'Angelo & Fortuin, 2021). This choice limits the number of particles in large scale problems, where it might be difficult to train and store several networks. Additionally, due to the interaction of all particles through the repulsion kernel, the training procedure is further complicated by requiring parallel training. Empirically, it has not been analyzed how parameter-efficient network structures perform as alternative parametric approximations. Thus, we propose to use a shared base network with multiple heads that represent the particles in function space, i.e. $f^{(i)}(\mathbf{x}; \theta_{\text{base}}, \theta^{(i)}_{\text{head}}) = f^{(i)}_{\text{head}}(f_{\text{base}}(\mathbf{x}; \theta_{\text{base}}); \theta^{(i)}_{\text{head}})$. By sharing the latent representation of the base model $f_{\text{base}}(\mathbf{x}; \theta_{\text{base}})$, our model is highly parameter-effective. We demonstrate in several experiments that an ensemble of linear layer is sufficient to improve uncertainty estimates of a single network.

Justification Multi-headed network architectures have been used successfully to distill DEs and replicate their functional behavior (Tran et al., 2020), demonstrating sufficient flexibility of a single shared network (Hinton et al., 2015). Performing particle optimization in function space mitigates the need for training a full DE prior to distillation. The use of a shared deterministic base network aligns with partially stochastic BNNs, where a subnetwork of the parameters is treated probabilistically. Most prominently, Bayesian last-layer networks are employed as practical means to reduce computational demands (Sharma et al., 2023). In App. B we provide arguments for the retrospective use of fs-RLL-E in pretrained networks.

Deep ensembles are not necessary: A single neural network with multiple heads is sufficiently flexible to provide diverse predictions for uncertainty estimation.

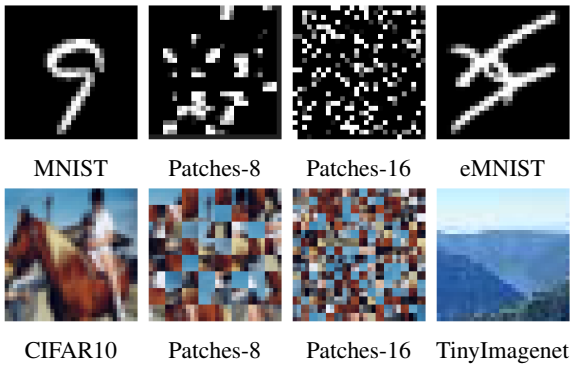


Figure 1. Example of repulsion samples for DirtyMNIST (top row) and CIFAR10/100 (bottom row).

3.2. Choice of repulsion evaluation samples

Evaluation of the function-space repulsion term requires selecting a set of *repulsion samples* $\mathbf{x}_{rep} \in \mathcal{D}_{\text{rep}}$. This choice significantly impacts the valid input domain for uncertainty estimates of fs-POVI methods. Prior work proposed to draw repulsion samples from the kernel density estimation (KDE) over training data (Wang et al., 2019), or take samples from the training data directly (D'Angelo & Fortuin, 2021). This choice, however, limits the applicability of the BNN approximation to situations where samples are coming from the same distribution as the training data. Selecting good repulsion samples becomes particularly challenging for high-dimensional spaces, for which drawing random samples from the entire input domain is simply infeasible. Instead, one must restrict the selection to an informative subset that covers the domain of interest from the input space. For image tasks, this often includes natural images from varying distributions. We can thus exploit the abundance of available unlabeled image data. For example, using eMNIST as repulsion samples for models trained on MNIST, or TinyImagenet for models trained on CIFAR10/100, leverages natural variability across different image sets. If unlabeled OOD data is unavailable, repulsion samples can be generated from the training data by label-destroying data augmentation techniques. One such effective method is the random shuffling of image patches, which destroys the shape information of objects that is crucial for human perception (see Figure 1).

Justification Enforcing diversity directly on the training data has been shown to degrade performance by artificially inflating epistemic uncertainty at data points where independent training would yield confident predictions (Abe et al., 2022; Jeffares et al., 2024). This approach often fails to detect OOD data, which may be characterized by spurious features present in the training data or features that are completely absent in the training set. Using unlabeled OOD data as repulsion samples provides an effective solution to

these challenges. These samples may contain features that are problematic or absent in the training data, allowing the models to meaningfully enforce diversity and improve OOD detection capabilities. Similarly, label-destroying data augmentation mitigates robust features that are indicative of the class label. By promoting diversity on those augmented images, it can prevent the model from depending on features that are not truly indicative of the class, thus mitigating the risk of overconfident and erroneous predictions. Compared to methods that rely on feature density to detect OOD data, repulsion samples offer the benefit of learning to ignore spurious features that may be present in the training data.

Encouraging diverse predictions on the training data itself is not sufficient to improve epistemic uncertainty estimation – we need random sampling, label-destroying data augmentation, or OOD data as repulsion samples.

4. Experiments

A single neural network (MAP) serves as the base model and backbone for all post-hoc uncertainty techniques. For unregularized DEs, we retrain the base network 5 times with random initializations (DE-5). We have selected two baselines for deterministic distance-based methods: DDU (Mukhoti et al., 2023) and SNGP (Liu et al., 2023). As a representative of single-mode Bayesian methods, we use the last-layer Laplace approximation (LL-Laplace) (Kristiadi et al., 2020). For our method, we compare the unregularized last-layer ensemble (LL-E), repulsion in parameter space (RLL-E), and repulsion in function space (fs-RLL-E) with varying repulsion samples. The LL-E consists of 10 particles with linear layers, introducing minimal computational overhead. To quantify epistemic uncertainty we use the softmax entropy (MAP, SNGP), mutual information (LL-Laplace, LL-E, DEs), and GMM density (DDU).

Common image classification datasets do not contain data points that are inherently ambiguous, i.e., data points that correspond to multiple classes. Even if aleatoric uncertainty (AU) and epistemic uncertainty (EU) are confounded, the evaluation would not reveal it. Thus, we use the DirtyMNIST dataset (Mukhoti et al., 2023) to evaluate the ability to distinguish ambiguous data (AU) and OOD data (EU). The DirtyMNIST dataset consists of clean MNIST digits and artificially generated ambiguous digits that belong to multiple classes. For OOD data, we use kMNIST (Clanuwat et al., 2018), Fashion MNIST (Xiao et al., 2017), and Omniglot (Lake et al., 2015).

Table 1 summarizes the ID performance and OOD detection gain we obtain by using the repulsive ensemble head with different choices of repulsion samples. Enforcing diversity of the parameters of the last layer particles (RLL-E) shows no improvement over the unregularized case (LL-E). Still, retraining the last layer increases accuracy, improving calibration, and enhancing OOD detection quality compared to the single network. Further improvements can be achieved with function space repulsion (fs-RLL-E), and an appropriate choice of repulsion samples (patches, eMNIST).

We present additional experimental results in the Appendix, specifically on disentangling aleatoric and epistemic uncertainty in active learning (App. C.3), further experiments on OOD detection (App. C.4), and on uncertainty calibration under distribution shifts (App. C.5).

5. Conclusion

We have shown that particle optimization in function space is not limited to DE architectures. A significant number of parameters can be saved by exploring different network architectures to parameterize the function space. We proposed a hybrid approach using a multi-headed network. The shared base network acts as a feature extractor for the repulsive ensemble head. This offers a principled way to provide already trained networks with retrospective uncertainty estimates, and to incorporate prior functional knowledge into the training procedure. Additionally, we highlighted the inherent limitations of enforcing diversity on training data alone. By utilizing augmented training data, or unlabeled OOD data, we achieved significant improvements on OOD detection without harming classification accuracy. We empirically demonstrate that our method successfully disentangles aleatoric and epistemic uncertainty, improves OOD detection, provides calibrated uncertainty estimates under distribution shifts, and performs well in active learning. At the same time, we significantly reduce the computational and memory requirements compared to DEs.

For future work, an important focus will be developing a rigorous relationship between the selection of the repulsion samples and the implications for the uncertainty estimates. We further aim to utilize data-augmentation schemes for generating task-specific repulsion samples.

Acknowledgments

The authors gratefully acknowledge the financial support under the scope of the COMET program within the K2 Center "Integrated Computational Material, Process and Product Engineering (IC-MPPE)" (Project No 886385). This program is supported by the Austrian Federal Ministries for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK) and for Labour and Economy (BMAW), represented by the Austrian Research Promotion Agency (FFG), and the federal states of Styria, Upper Austria and Tyrol.

Table 1. Comparison of uncertainty decomposition on DirtyMNIST. Aleatoric uncertainty (AU), and epistemic uncertainty (EU) are used to detect ambiguous, and OOD samples. Mean and standard deviation are computed over 10 runs. Best results are in bold, second best are underlined.

Method	ACC. ↑ [%]	NLL ↓ [%]	ECE ↓ [%]	OOD AUROC ↑ [%]		
				MNIST vs ambig. (AU)	MNIST vs. OOD (EU)	ambig. vs OOD (EU)
MAP	$79.97_{\pm 0.77}$	$58.12_{\pm 1.77}$	$2.82_{\pm 0.65}$	$93.83_{\pm 0.7}$	$97.71_{\pm 0.65}$	$76.11_{\pm 4.75}$
DDU	$79.97_{\pm 0.77}$	$58.12_{\pm 1.77}$	$2.82_{\pm 0.65}$	$93.83_{\pm 0.7}$	$\mathbf{99.78_{\pm 0.02}}$	$\mathbf{99.96_{\pm 0.01}}$
SNGP	$\underline{83.49_{\pm 0.11}}$	$49.78_{\pm 0.13}$	$3.98_{\pm 0.09}$	$88.76_{\pm 0.39}$	$94.68_{\pm 1.68}$	$73.88_{\pm 4.49}$
LL-Laplace	$80.73_{\pm 1.30}$	$55.90_{\pm 2.78}$	$2.03_{\pm 0.57}$	$94.3_{\pm 1.6}$	$98.41_{\pm 0.46}$	$93.15_{\pm 2.64}$
LL-E *(ours)*	$\mathbf{83.53_{\pm 0.16}}$	$\mathbf{48.32_{\pm 0.24}}$	$\mathbf{1.00_{\pm 0.14}}$	$\mathbf{96.82_{\pm 0.34}}$	$99.41_{\pm 0.22}$	$96.16_{\pm 1.53}$
RLL-E *(ours)*	$\mathbf{83.53_{\pm 0.16}}$	$\mathbf{48.32_{\pm 0.24}}$	$\mathbf{1.00_{\pm 0.14}}$	$\mathbf{96.82_{\pm 0.34}}$	$99.41_{\pm 0.22}$	$96.16_{\pm 1.53}$
fs-RLL-E *(ours)*						
+ *dirtyMNIST*	$83.24_{\pm 0.20}$	$49.21_{\pm 0.29}$	$1.18_{\pm 0.12}$	$96.38_{\pm 0.34}$	$99.29_{\pm 0.43}$	$95.36_{\pm 2.82}$
+ *eMNIST*	$\underline{83.52_{\pm 0.20}}$	$48.91_{\pm 0.24}$	$1.18_{\pm 0.20}$	$95.27_{\pm 2.07}$	$99.3_{\pm 0.3}$	$\underline{99.52_{\pm 0.23}}$
+ *Patches-16*	$\underline{83.51_{\pm 0.16}}$	$\underline{48.35_{\pm 0.24}}$	$\underline{1.03_{\pm 0.13}}$	$\underline{96.74_{\pm 2.14}}$	$\underline{99.52_{\pm 0.26}}$	$97.69_{\pm 1.38}$
+ *Patches-8*	$\underline{83.52_{\pm 0.15}}$	$48.40_{\pm 0.24}$	$\underline{1.02_{\pm 0.15}}$	$96.63_{\pm 1.81}$	$99.4_{\pm 0.25}$	$98.59_{\pm 0.61}$
+ *Patches-4*	$\underline{83.50_{\pm 0.18}}$	$48.59_{\pm 0.23}$	$\underline{1.05_{\pm 0.17}}$	$96.44_{\pm 2.0}$	$99.45_{\pm 0.21}$	$99.09_{\pm 0.38}$
DE-5	$83.31_{\pm 0.15}$	$50.26_{\pm 0.40}$	$5.01_{\pm 0.67}$	$96.23_{\pm 0.15}$	$98.96_{\pm 0.2}$	$93.88_{\pm 1.57}$

References

Abe, T., Buchanan, E. K., Pleiss, G., and Cunningham, J. P. The best deep ensembles sacrifice predictive diversity. 2022.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural network. pp. 1613–1622. PMLR, 2015.

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

Charpentier, B., Zügner, D., and Günnemann, S. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. *Advances in Neural Information Processing Systems*, 33:1356–1367, 2020.

Chen, C., Zhang, R., Wang, W., Li, B., and Chen, L. A unified particle-optimization framework for scalable bayesian sampling. *arXiv preprint arXiv:1805.11659*, 2018.

Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., and Ha, D. Deep learning for classical japanese literature, 2018.

D'Angelo, F. and Fortuin, V. Repulsive deep ensembles are Bayesian. *Advances in Neural Information Processing Systems*, 34:3451–3465, 2021.

Daxberger, E., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M., and Hennig, P. Laplace redux-effortless Bayesian deep learning. *Advances in Neural Information Processing Systems*, 34:20089–20103, 2021.

Depeweg, S., Hernandez-Lobato, J.-M., Doshi-Velez, F., and Udluft, S. Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning. pp. 1184–1193. PMLR, 2018.

Dusenberry, M., Jerfel, G., Wen, Y., Ma, Y., Snoek, J., Heller, K., Lakshminarayanan, B., and Tran, D. Efficient and scalable Bayesian neural nets with rank-1 factors. pp. 2782–2792. PMLR, 2020.

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. pp. 1050–1059. PMLR, 2016.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein GANs. *Advances in Neural Information Processing Systems*, 30, 2017.

Gustafsson, F. K., Danelljan, M., and Schon, T. B. Evaluating scalable bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 318–319, 2020.

Harrison, J., Willes, J., and Snoek, J. Variational bayesian last layers. *arXiv preprint arXiv:2404.11599*, 2024.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. *International Conference on Learning Representations*, 2019.

Hendrycks, D., Mazeika, M., and Dietterich, T. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*,

2019. URL https://openreview.net/forum?id=HyxCxhRcY7.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Jeffares, A., Liu, T., Crabbé, J., and van der Schaar, M. Joint training of deep ensembles fails due to learner collusion. *Advances in Neural Information Processing Systems*, 36, 2024.

Kirsch, A. Bridging the data processing inequality and function-space variational inference. In *The Third Blogpost Track at ICLR 2024*, 2024. URL https://openreview.net/forum?id=CAt2LeiiSc.

Kristiadi, A., Hein, M., and Hennig, P. Being Bayesian, even just a bit, fixes overconfidence in relu networks. pp. 5436–5446. PMLR, 2020.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017.

Lee, S., Purushwalkam, S., Cogswell, M., Crandall, D., and Batra, D. Why m heads are better than one: Training a diverse ensemble of deep networks. *arXiv preprint arXiv:1511.06314*, 2015.

Liu, C., Zhuo, J., Cheng, P., Zhang, R., and Zhu, J. Understanding and accelerating particle-based variational inference. pp. 4082–4092. PMLR, 2019.

Liu, J., Lin, Z., Padhy, S., Tran, D., Bedrax Weiss, T., and Lakshminarayanan, B. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, 33:7498–7512, 2020.

Liu, J. Z., Padhy, S., Ren, J., Lin, Z., Wen, Y., Jerfel, G., Nado, Z., Snoek, J., Tran, D., and Lakshminarayanan, B. A simple approach to improve single-model deep uncertainty via distance-awareness. *J. Mach. Learn. Res.*, 24:42–1, 2023.

Liu, Q. Stein variational gradient descent as gradient flow. *Advances in Neural Information Processing Systems*, 30, 2017.

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

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

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

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Mukhoti, J., Kirsch, A., van Amersfoort, J., Torr, P. H., and Gal, Y. Deep deterministic uncertainty: A new simple baseline. pp. 24384–24394, 2023.

Neal, R. M. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. Deep exploration via bootstrapped dqn. *Advances in Neural Information Processing Systems*, 29, 2016.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems*, 32, 2019.

Postels, J., Blum, H., Cadena, C., Siegwart, R., Van Gool, L., and Tombari, F. Quantifying aleatoric and epistemic uncertainty using density estimation in latent space. *arXiv preprint arXiv:2012.03082*, 1, 2020.

Postels, J., Segu, M., Sun, T., Sieber, L., Van Gool, L., Yu, F., and Tombari, F. On the practicality of deterministic epistemic uncertainty. *arXiv preprint arXiv:2107.00649*, 2021.

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

Rudner, T. G., Kapoor, S., Qiu, S., and Wilson, A. G. Function-space regularization in neural networks: A probabilistic perspective. *International Conference on Machine Learning*, 2023.

Schweighofer, K., Aichberger, L., Ielanskyi, M., Klambauer, G., and Hochreiter, S. Quantification of uncertainty with adversarial models. *Advances in Neural Information Processing Systems*, 36:19446–19484, 2023.

Sercu, T., Puhrsch, C., Kingsbury, B., and LeCun, Y. Very deep multilingual convolutional neural networks for lvcsr. pp. 4955–4959. IEEE, 2016.

Sharma, M., Farquhar, S., Nalisnick, E., and Rainforth, T. Do Bayesian neural networks need to be fully stochastic? pp. 7694–7722. PMLR, 2023.

Song, G. and Chai, W. Collaborative learning for deep neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

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

Tagasovska, N. and Lopez-Paz, D. Single-model uncertainties for deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Tran, L., Veeling, B. S., Roth, K., Swiatkowski, J., Dillon, J. V., Snoek, J., Mandt, S., Salimans, T., Nowozin, S., and Jenatton, R. Hydra: Preserving ensemble diversity for model distillation. *arXiv preprint arXiv:2001.04694*, 2020.

Trinh, T., Heinonen, M., Acerbi, L., and Kaski, S. Input gradient diversity for neural network ensembles. *arXiv preprint arXiv:2306.02775*, 2023.

Valdenegro-Toro, M. Sub-ensembles for fast uncertainty estimation in neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4119–4127, 2023.

van Amersfoort, J., Smith, L., Teh, Y. W., and Gal, Y. Uncertainty estimation using a single deep deterministic neural network. pp. 9690–9700. PMLR, 2020.

van Amersfoort, J., Smith, L., Jesson, A., Key, O., and Gal, Y. On feature collapse and deep kernel learning for single forward pass uncertainty. *arXiv preprint arXiv:2102.11409*, 2021.

Wang, Z., Ren, T., Zhu, J., and Zhang, B. Function space particle optimization for Bayesian neural networks. *arXiv preprint arXiv:1902.09754*, 2019.

Wild, V. D., Ghalebikesabi, S., Sejdinovic, D., and Knoblauch, J. A rigorous link between deep ensembles and (variational) bayesian methods. In *Advances in Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=eTHawKFT4h.

Wilson, A. G. The case for bayesian deep learning. *arXiv preprint arXiv:2001.10995*, 2020.

Wimmer, L., Sale, Y., Hofman, P., Bischl, B., and Hüllermeier, E. Quantifying aleatoric and epistemic uncertainty in machine learning: Are conditional entropy and mutual information appropriate measures? In *Uncertainty in Artificial Intelligence*, pp. 2282–2292. PMLR, 2023.

Winkens, J., Bunel, R., Roy, A. G., Stanforth, R., Natarajan, V., Ledsam, J. R., MacWilliams, P., Kohli, P., Karthikesalingam, A., Kohl, S., et al. Contrastive training for improved out-of-distribution detection. *arXiv preprint arXiv:2007.05566*, 2020.

Xia, G. and Bouganis, C.-S. On the usefulness of deep ensemble diversity for out-of-distribution detection. *arXiv preprint arXiv:2207.07517*, 2022.

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

Yashima, S., Suzuki, T., Ishikawa, K., Sato, I., and Kawakami, R. Feature space particle inference for neural network ensembles. pp. 25452–25468. PMLR, 2022.

Zhu, X., Gong, S., et al. Knowledge distillation by on-the-fly native ensemble. *Advances in Neural Information Processing Systems*, 31, 2018.

A. Why repulsion matters for a finite number of particles

Importantly, the guarantee of convergence to the posterior distribution $p(\theta|\mathcal{D})$ is only valid in the limit of infinite number of particles. Although fascinating from a theoretical perspective, practical importance lies in the analysis of the behavior for a finite number of particles. In this section, we discuss the importance of the repulsion term in the regime where the number of particles is significantly smaller than the number of local minima.

Estimating epistemic uncertainty

Uncertainty estimates in model predictions are typically derived from the disagreement among ensemble members. Following (Depeweg et al., 2018; Wimmer et al., 2023; Schweighofer et al., 2023), this uncertainty can be decomposed into aleatoric and epistemic components, represented as conditional entropy and mutual information, respectively:

$$\underbrace{\mathbb{H}[\mathbb{E}_{p(\theta|\mathcal{D})}[p(\mathbf{y}|\mathbf{x},\theta)]]}_{\textit{Total}} = \underbrace{\mathbb{E}_{p(\theta|\mathcal{D})}[\mathbb{H}[p(\mathbf{y}|\mathbf{x},\theta)]]}_{\textit{Aleatoric}} + \underbrace{\mathbb{I}[\mathbf{y};\theta|\mathbf{x},\mathcal{D}]}_{\textit{Epistemic}} \tag{4}$$

The total uncertainty is given by the entropy of the model's predictions, aleatoric uncertainty represents the variability in outcomes due to inherent randomness in the data, and epistemic uncertainty reflects our lack of knowledge about which model generated the data. The mutual information, obtained by integrating over the Kullback-Leibler divergence, estimates the expected epistemic uncertainty:

$$\mathbb{I}[\mathbf{y};\theta|\mathbf{x},\mathcal{D}] = \mathbb{E}_{p(\theta|\mathcal{D})}\left[D_{\mathrm{KL}}\left(p(\mathbf{y}|\mathbf{x},\theta) \,||\, p(\mathbf{y}|\mathbf{x},\mathcal{D})\right)\right] \tag{5}$$

If a test sample $\mathbf{x}$ can be explained by many disagreeing models $p(\mathbf{y}|\mathbf{x},\theta)$, each plausible under the posterior distribution $p(\theta|\mathcal{D})$, epistemic uncertainty is high. By acquiring additional training data close to $\mathbf{x}$, the space of plausible models and thus inconsistent predictions is decreased.

Finite particle approximation

If the posterior distribution is approximated by a finite set of discrete particles $\theta^{(i)}$, epistemic uncertainty estimation simplifies to a Monte Carlo integration (Wimmer et al., 2023):

$$\frac{1}{n}\sum_{i=1}^{n} D_{\mathrm{KL}}\left(p\left(\mathbf{y}|\mathbf{x},\theta^{(i)}\right) \Big|\Big| \frac{1}{n}\sum_{j=1}^{n} p\left(\mathbf{y}|\mathbf{x},\theta^{(j)}\right)\right) \tag{6}$$

Given practical constraints on the number of particles (typically five to ten), many posterior modes remain unexplored. The estimate of the epistemic uncertainty is shaped largely by a very limited number of posterior modes. This limitation stresses the need for guiding particles towards representative and *diverse posterior modes* to avoid underestimation of epistemic uncertainty.

Repulsion in finite deep ensembles

Deep ensembles can be viewed as an unregularized case of Equation (2), lacking a repulsion term. Particles move according to the gradient flow towards high-density posterior modes $p(\theta_l^{(j)}|\mathcal{D})$, with diversity stemming from their random initial positions $\theta_{l=0}^{(j)}$ in the loss landscape. Recent research has raised concerns about the effectiveness of this approach in achieving diverse posterior modes. The loss landscape, heavily influenced by input features correlated with the target, may render diverse posterior modes inaccessible to the unregularized gradient flow (Schweighofer et al., 2023). In addition, empirical evidence has demonstrated that the epistemic uncertainty provided by DEs does not reliably identify distribution shifts. In several cases, the aleatoric uncertainty of a single model has been more effective in detecting OOD data (Schweighofer et al., 2023; Xia & Bouganis, 2022).

Repulsive deep ensembles (RDEs) introduce a repulsion kernel, $k\left(\theta^{(i)},\theta^{(j)}\right)$, to prevent particles from converging to identical posterior modes. Theoretically, when the number of particles approaches infinity, this repulsion mechanism ensures convergence to the true posterior distribution (D'Angelo & Fortuin, 2021; Wild et al., 2023). In practical applications, where the number of particles is finite and vastly smaller than the number of local optima, studies show that random initialization is sufficient to prevent the particles from collapsing into the same local optimum (Wild et al., 2023). Still, it is not guaranteed that those distinct local optima result in diverse prediction functions.

Consequently, to improve performance of unregularized DEs, the repulsion kernel needs to actively direct particles towards distinct and diverse posterior modes. Specifically, in this work we aim to meet the following desiderata:

D1 *The repulsion term should steer particles towards diverse posterior modes, which provide a useful approximation for the epistemic uncertainty in Equation* (5)*.*

D2 *Particles should reach diverse posterior modes from the same initial parameters through the use of the repulsion term. This enables the fine-tuning of pre-trained models to better approximate epistemic uncertainty.*

B. Retrospective uncertainties for pre-trained models

The multi-headed network approach provides a principled approach to computing retrospective uncertainties for a pre-trained base network. We replace the last layer of the base network with an ensemble of linear layers trained with the function space repulsion term. In this way, representation learning and function space inference are decoupled, allowing the computation of diverse decision boundaries while leveraging pre-learned representations. If the function space repulsion term is included post hoc, features indicative of OOD data may not be extracted by the base network due to feature collapse, where data points far apart in input space collapse into indistinguishable parts in feature space (van Amersfoort et al., 2020). Fortunately, pre-trained deep neural networks are often trained with mechanisms that help mitigate feature collapse. In the following, we discuss important techniques commonly used in training deep networks that lead to feature space regularization of the learned representations. In Section 4, we evaluate whether the last-layer retraining is sufficiently flexible to enforce diverse predictions on OOD data. This directly addresses our second goal (D2) of obtaining diverse posterior modes from the same initial parameters.

Spectral normalization and residual connections Distance-aware representations can be achieved by imposing bi-Lipschitz constraints $K_L d_I(\mathbf{x}_1, \mathbf{x}_2) \leq d_F(f_{\text{base}}(\mathbf{x}_1), f_{\text{base}}(\mathbf{x}_2)) \leq K_U d_I(\mathbf{x}_1, \mathbf{x}_2)$. Here, d_I and d_F represent distance measures in the input and feature spaces, while K_L and K_U are the lower and upper Lipschitz constants, respectively. These constraints enforce a bounded relationship between distances in the input (d_I) and feature (d_F) space. Models with constrained Lipschitz constants have demonstrated improved generalization and adversarial risk mitigation (Miyato et al., 2018). Spectral normalization and residual connections serve as effective techniques to impose these constraints (Miyato et al., 2018). They prevent feature collapse while introducing smoothness (upper Lipschitz constant) in the feature space. While most pretrained models are not trained with spectral normalization, employing a network structure with residual connections alone often suffices to maintain distance awareness in feature space.

Data augmentation Data augmentation is an important training technique that aims to enrich the data set by generating different variations of the original samples. Techniques such as Mixup or CutMix are often used for this purpose. They introduce variations by combining or interpolating between different samples, thereby expanding the model's exposure to a broader range of data distributions. Suppose we train a classifier to distinguish between the digits "0" and "1". Initially, the model might rely on simple features, such as the presence or absence of a straight line, to make accurate predictions. While this approach may be sufficient for distinguishing between "0" and "1", it may struggle when faced with more complex tasks, such as identifying the digit "7" as OOD. This difficulty stems from the model's limited exposure to diverse features during training. Next, consider CutMix augmentation, where parts of different images are combined to create new synthetic samples. For example, by merging the top half of a "0" image with the bottom half of a "1" image, we create a synthetic sample that resembles a "7". Incorporating such augmented samples into the training data forces the model to learn more nuanced features that distinguish not only between "0" and "1", but also between "7" and the other digits. This process enriches the model's representation of the data and promotes the extraction of more diverse features.

Pre-trained neural networks are often trained with methods that avoid feature collapse. We can decouple the problem into two stages: representation learning and uncertainty-aware fine-tuning using function-space inference.

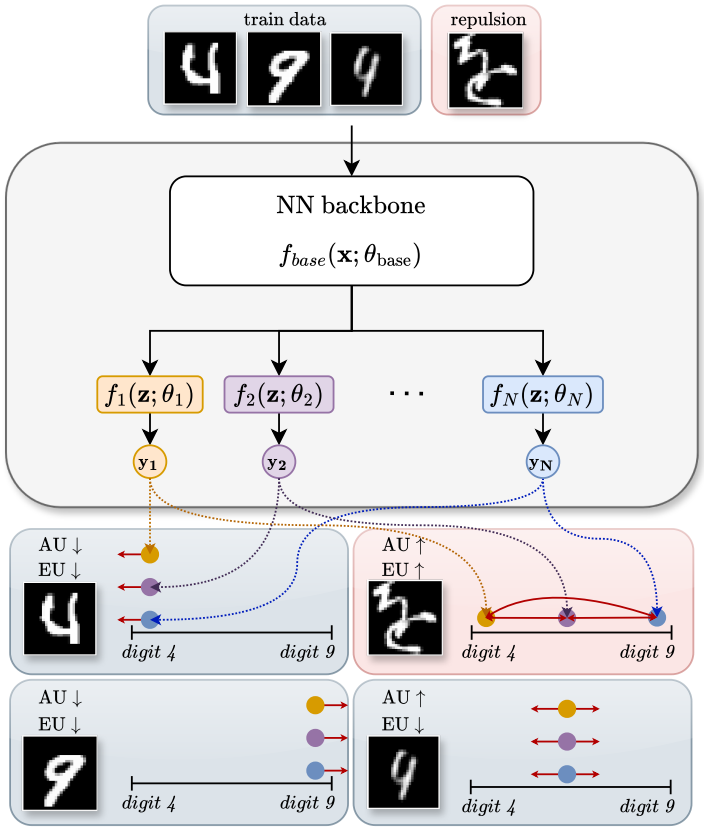


Figure 2. Function-space RLL-E (fs-RLL-E), with N particles. Colored dots correspond to the prediction of a particle. Unlabeled data points from a different distribution are used as repulsion samples for the function space repulsion loss. Epistemic uncertainty (EU) is the lowest, when all particle predictions agree, and increases with the spread of the particles. The aleatoric uncertainty (AU) increases with ambiguous samples, e.g. the digit on the lower right belonging to both classes, resulting in particle predictions centered in the probability region.

C. Additional experimental results

C.1. Training details

For particle-based inference in function space (Wang et al., 2019; D'Angelo & Fortuin, 2021), we relied on the implementation available at `https://github.com/ratschlab/repulsive_ensembles`, and for DDU (Mukhoti et al., 2023) at `https://github.com/omegafragger/DDU`. Table 2 summarizes relevant hyperparameters for training the base networks and the repulsive ensemble head.

Spectral normalization We follow the implementation of (Mukhoti et al., 2023) and utilize base networks with residual connections and spectral normalization. As proposed in SNGP and DDU (Liu et al., 2023; Mukhoti et al., 2023), spectral normalization is used to bound the Lipschitz constant of the network. Online spectral normalization with a one step power iteration is applied to convolutional weights, and exact spectral normalization is applied to 1x1 convolutional layers (Mukhoti et al., 2023).

Computational overhead In all experiments, we use a pretrained base network as the feature extractor for the uncertainty estimation. For the image classification experiments, we do not modify the base network and add an ensemble head consisting of linear layers only. Thus, the number of trainable parameters of our method is determined by the dimension of the feature space of the base network d, the number of classes K, and the number of particles n, i.e., $(d \times K + K) \times n$. The feature space dimension for various base networks is shown in Table 3.

Table 2. Implementation details and hyperparameter for the different experiments.

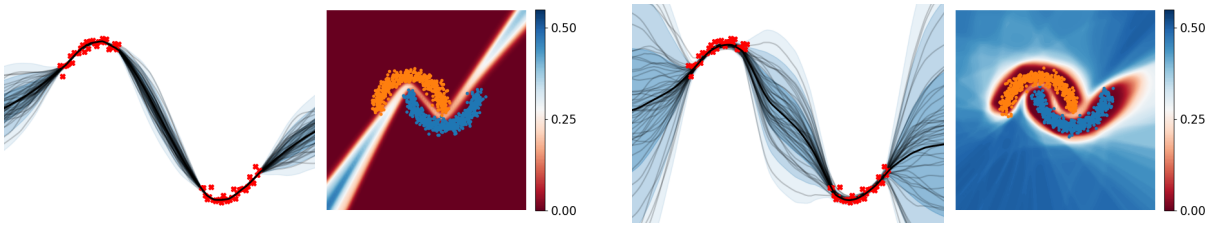
TASK	ARCHITECTURE	HYPERPARAMETER	VALUE
IMAGE CLASSIFICATION BASE NETWORK	RESNET-18 WIDE-RESNET-28-10	EPOCHS	50 (DirtyMNIST) 300 (CIFAR10/100)
		OPTIMIZER	SGD
		LEARNING RATE	0.1 0.01 – epoch 25 (DirtyMNIST), epoch 150 (CIFAR10/100) 0.001 – epoch 40 (DirtyMNIST), epoch 250 (CIFAR10/100)
		MOMENTUM	0.9
ACTIVE LEARNING BASE NETWORK	RESNET-18	EPOCHS	20
		OPTIMIZER	Adam
		LEARNING RATE	0.001
LAST-LAYER-ENSEMBLE	FULLY CONNECTED	EPOCHS	30
		# HIDDEN LAYER # NEURONS PER LAYER	0 10
		LEARNING RATE	0.0001
		# BATCH SIZE TRAINING DATA # BATCH SIZE REPULSION SAMPLES	128 128

Table 3. Feature space dimension of different base network architectures.

LENET	$d = 84$
VGG-16	$d = 512$
RESNET-18	$d = 512$
WIDERESNET-28-10	$d = 640$
RESNET-50	$d = 2048$
RESNET-101	$d = 2048$

C.2. Synthetic Data

On two toy examples, we illustrate the effectiveness of the multi-head architecture as a lightweight parameterization and the advantages of performing inference in function space. We estimate the epistemic uncertainty for a one-dimensional regression and a two-dimensional classification problem using full DEs and fs-RLL-E. A feed-forward neural network with 3 hidden layers and 128 neurons is used as the base network. The repulsive head consists of 30 particles with linear layers. Results are shown in Figure 3. Deep ensemble predictions show low uncertainty far from the training data. By performing particle inference in function space, we can enforce diverse predictions outside the training distribution even with a simpler network structure.



(a) Unregularized DEs (b) Function space RLL-E

Figure 3. Predictions of DEs and the proposed fs-RLL-E. For regression, we show the prediction of individual particles, the mean and the standard deviation. For classification on the two-moons data, we show the standard deviation of the predicted probabilities $p(\mathbf{y}|\mathbf{x}, \theta)$. All DE members learn the same decision boundary and are thus highly confident in regions distant from training data, while fs-RLL-E predictions are enforced to be diverse outside of the training data.

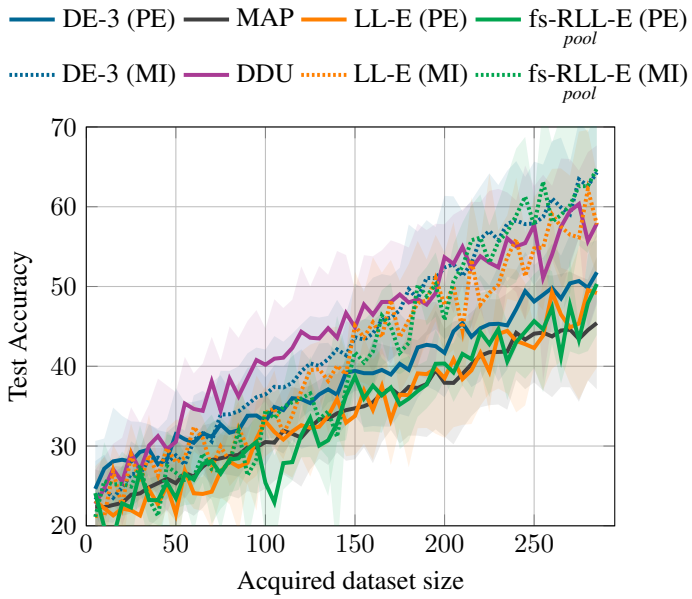


Figure 4. Test accuracy of the model as a function of the data samples that are acquired using the different uncertainty estimates. Using the mutual information (MI) of the LL-E and fs-RLL-E prediction outperforms softmax entropy of the single network and performs on par with the other uncertainty baselines.

C.3. Uncertainty decomposition for active learning

We evaluate the performance of our LL-E and fs-RLL-E uncertainty estimates on an active learning task proposed in (Mukhoti et al., 2023). Given a small number of initial training points and a large pool of unlabeled data, the aim is to select the most informative data points, which are subsequently used to retrain the network. We report the results using softmax entropy of a single Resnet-18, DDU density, mutual information (MI) and predictive entropy (PE) of an ensemble of 3 Resnet-18, LL-E, and fs-RLL-E ($\mathbf{x}_C = \text{POOL}$). We start with an initial training set of 20 samples and a pool of clean and ambiguous MNIST samples. The ratio of clean to ambiguous is 1:60. In each iteration, we add the 5 samples with the highest epistemic uncertainty in the pool set. Thus, disentangling aleatoric and epistemic uncertainty is essential to select informative samples that improve prediction accuracy. Fig. 4 shows that DDU, LL-E and fs-RLL-E are all able to compete with DEs. All methods achieve a similar accuracy on the test set at the end of the iterations. We clearly see the importance of using epistemic uncertainty to avoid selecting ambiguous samples with high aleatoric uncertainty. The results are averaged over 5 runs with different random seed.

C.4. Semantic shift detection

To further verify the epistemic uncertainty estimates of our method, we perform OOD detection on larger image classification tasks. The results are summarized in Tables 4 and 5. We train a Wide-Resnet-28-10 for CIFAR10 and CIFAR100. Again, our LL-E head consists of 10 particles with linear layers only.
In terms of accuracy and in-distribution (ID) calibration, DEs perform the best, followed by our function-space RLL-E (fs-RLL-E). As we only retrain the last layer of the base network, we do not expect to improve the prediction accuracy. We emphasize that our aim is to introduce computationally cheap uncertainty estimates that are informative about erroneous predictions and OOD data. For both CIFAR10 and CIFAR100, retraining the last layer leads to improvements over the base network (MAP) for all examined uncertainty scores. While last-layer retraining does not lead to improved accuracy, NLL and ECE are consistently improved, which indicates better calibration of the uncertainty estimates. Also, function space diversity on augmented training data can lead to further improvements for ID uncertainty estimates.

Table 4. Comparison of uncertainty estimation for ID calibration, and OOD detection on CIFAR10. Mean and standard deviation are computed over 10 runs. Best results are in bold, second best are underlined.

Method	ACC. ↑ [%]	NLL ↓ [%]	ECE ↓ [%]	OOD AUROC ↑ [%]					
				Cifar100	*TinyIm.*	*Places365*	*Texture*	*SVHN*	*FakeData*
MAP	$95.91_{\pm 0.13}$	$15.94_{\pm 0.61}$	$2.33_{\pm 0.13}$	$89.71_{\pm 0.26}$	$89.25_{\pm 0.26}$	$90.00_{\pm 0.35}$	$89.21_{\pm 0.90}$	$93.46_{\pm 2.54}$	$97.06_{\pm 3.60}$
DDU	$95.91_{\pm 0.13}$	$15.94_{\pm 0.61}$	$2.33_{\pm 0.13}$	$89.00_{\pm 0.46}$	$89.14_{\pm 0.43}$	$90.81_{\pm 0.38}$	$\mathbf{97.62_{\pm 0.30}}$	$\mathbf{98.62_{\pm 0.37}}$	$\mathbf{100.00_{\pm 0.00}}$
SNGP	$95.90_{\pm 0.13}$	$13.66_{\pm 0.46}$	$1.25_{\pm 0.12}$	$89.44_{\pm 0.26}$	$88.73_{\pm 0.30}$	$\mathbf{91.85_{\pm 0.39}}$	$95.30_{\pm 0.48}$	$93.15_{\pm 3.09}$	$97.13_{\pm 3.91}$
LL-Laplace	$95.89_{\pm 0.10}$	$14.39_{\pm 0.47}$	$1.13_{\pm 0.10}$	$88.77_{\pm 0.29}$	$88.57_{\pm 0.24}$	$89.63_{\pm 0.49}$	$89.15_{\pm 0.70}$	$92.30_{\pm 1.82}$	$96.74_{\pm 3.26}$
LL-E *(ours)*	$95.87_{\pm 0.15}$	$13.64_{\pm 0.41}$	$0.97_{\pm 0.20}$	$89.80_{\pm 0.37}$	$89.73_{\pm 0.40}$	$91.06_{\pm 0.53}$	$89.90_{\pm 0.91}$	$92.56_{\pm 2.51}$	$97.97_{\pm 2.39}$
RLL-E *(ours)*	$95.88_{\pm 0.15}$	$13.68_{\pm 0.41}$	$0.97_{\pm 0.20}$	$89.74_{\pm 0.33}$	$89.65_{\pm 0.37}$	$91.03_{\pm 0.43}$	$89.87_{\pm 0.88}$	$92.40_{\pm 2.51}$	$97.79_{\pm 2.53}$
fs-RLL-E *(ours)*									
+ *Cifar10*	$95.87_{\pm 0.16}$	$19.08_{\pm 0.56}$	$5.73_{\pm 0.25}$	$45.16_{\pm 6.63}$	$44.78_{\pm 6.43}$	$44.67_{\pm 9.05}$	$42.01_{\pm 11.49}$	$39.41_{\pm 13.19}$	$35.12_{\pm 21.10}$
+ *Cifar100*	$95.78_{\pm 0.12}$	$17.25_{\pm 0.33}$	$3.69_{\pm 0.27}$	$\mathbf{91.93_{\pm 0.22}}$	$\mathbf{91.46_{\pm 0.23}}$	$91.45_{\pm 0.41}$	$94.06_{\pm 0.53}$	$96.34_{\pm 1.24}$	$98.45_{\pm 1.05}$
+ *TinyImagenet*	$95.83_{\pm 0.14}$	$15.78_{\pm 0.47}$	$1.58_{\pm 0.29}$	$89.47_{\pm 0.97}$	$90.40_{\pm 0.46}$	$91.64_{\pm 0.48}$	$94.23_{\pm 1.15}$	$94.22_{\pm 2.42}$	$99.58_{\pm 0.41}$
+ *Texture*	$95.82_{\pm 0.15}$	$15.91_{\pm 0.39}$	$1.97_{\pm 0.27}$	$89.48_{\pm 0.59}$	$89.92_{\pm 0.48}$	$89.48_{\pm 0.54}$	$95.60_{\pm 0.37}$	$95.54_{\pm 1.51}$	$98.51_{\pm 1.84}$
+ *Patches-32*	$95.86_{\pm 0.15}$	$14.01_{\pm 0.40}$	$0.66_{\pm 0.14}$	$86.54_{\pm 1.31}$	$86.39_{\pm 1.63}$	$84.71_{\pm 2.83}$	$90.80_{\pm 3.18}$	$91.08_{\pm 5.05}$	$99.69_{\pm 0.93}$
+ *Patches-16*	$95.88_{\pm 0.15}$	$14.33_{\pm 0.40}$	$\mathbf{0.62_{\pm 0.14}}$	$82.11_{\pm 1.62}$	$84.22_{\pm 1.50}$	$85.89_{\pm 1.73}$	$87.36_{\pm 2.32}$	$77.91_{\pm 6.26}$	$98.42_{\pm 2.11}$
DE-5	$\mathbf{96.55_{\pm 0.08}}$	$\mathbf{11.17_{\pm 0.11}}$	$0.81_{\pm 0.06}$	$91.59_{\pm 0.07}$	$90.63_{\pm 0.08}$	$90.93_{\pm 0.14}$	$94.20_{\pm 0.25}$	$96.95_{\pm 0.42}$	$99.75_{\pm 0.41}$

Table 5. Comparison of uncertainty estimation for ID calibration, and OOD detection on CIFAR100. Mean and standard deviation are computed over 10 runs. Best results are in bold, second best are underlined.

Method	ACC. ↑ [%]	NLL ↓ [%]	ECE ↓ [%]	OOD AUROC ↑ [%]				
				TinyIm.	*Places365*	*Texture*	*SVHN*	*FakeData*
MAP	$80.84_{\pm 0.19}$	$79.31_{\pm 0.97}$	$6.83_{\pm 0.22}$	$82.78_{\pm 0.15}$	$79.41_{\pm 0.16}$	$78.52_{\pm 0.79}$	$84.00_{\pm 1.40}$	$82.05_{\pm 15.51}$
DDU	$80.84_{\pm 0.19}$	$79.31_{\pm 0.97}$	$6.83_{\pm 0.22}$	$54.14_{\pm 2.55}$	$59.59_{\pm 1.90}$	$80.64_{\pm 2.29}$	$70.67_{\pm 2.38}$	$99.94_{\pm 0.13}$
SNGP	$80.65_{\pm 0.20}$	$83.18_{\pm 1.16}$	$9.51_{\pm 0.33}$	$79.62_{\pm 1.14}$	$\mathbf{85.77_{\pm 1.80}}$	$96.72_{\pm 0.93}$	$73.95_{\pm 2.57}$	$68.48_{\pm 24.75}$
LL-Laplace	$80.47_{\pm 0.27}$	$81.04_{\pm 1.68}$	$6.49_{\pm 0.36}$	$\mathbf{83.29_{\pm 0.35}}$	$79.97_{\pm 0.41}$	$81.44_{\pm 0.67}$	$81.57_{\pm 1.69}$	$84.84_{\pm 17.10}$
LL-E *(ours)*	$80.58_{\pm 0.20}$	$75.83_{\pm 0.72}$	$3.64_{\pm 0.25}$	$82.54_{\pm 0.19}$	$78.77_{\pm 0.35}$	$85.55_{\pm 0.77}$	$85.86_{\pm 1.90}$	$90.25_{\pm 20.87}$
RLL-E *(ours)*	$80.55_{\pm 0.21}$	$75.94_{\pm 0.72}$	$3.60_{\pm 0.26}$	$82.55_{\pm 0.18}$	$78.79_{\pm 0.36}$	$85.65_{\pm 0.83}$	$85.85_{\pm 1.76}$	$91.63_{\pm 19.52}$
fs-RLL-E *(ours)*								
+ *Cifar100*	$80.56_{\pm 0.19}$	$78.28_{\pm 0.69}$	$6.38_{\pm 0.32}$	$57.05_{\pm 2.96}$	$57.65_{\pm 3.99}$	$51.52_{\pm 4.49}$	$45.82_{\pm 6.93}$	$50.75_{\pm 18.87}$
+ *TinyImagenet*	$80.51_{\pm 0.19}$	$77.77_{\pm 0.76}$	$4.05_{\pm 0.23}$	$82.61_{\pm 0.56}$	$82.35_{\pm 1.06}$	$90.40_{\pm 0.99}$	$94.34_{\pm 1.60}$	$99.94_{\pm 0.08}$
+ *Texture*	$80.33_{\pm 0.16}$	$80.14_{\pm 0.83}$	$4.55_{\pm 0.27}$	$79.29_{\pm 0.61}$	$76.49_{\pm 1.07}$	$\mathbf{97.13_{\pm 0.20}}$	$\mathbf{96.24_{\pm 1.43}}$	$99.60_{\pm 0.30}$
+ *Patches-32*	$80.59_{\pm 0.18}$	$75.85_{\pm 0.73}$	$3.63_{\pm 0.26}$	$81.51_{\pm 0.46}$	$76.95_{\pm 0.63}$	$87.98_{\pm 1.64}$	$86.08_{\pm 2.80}$	$\mathbf{100.00_{\pm 0.00}}$
+ *Patches-16*	$80.59_{\pm 0.17}$	$75.98_{\pm 0.71}$	$3.68_{\pm 0.28}$	$81.22_{\pm 0.48}$	$81.34_{\pm 1.18}$	$90.15_{\pm 1.13}$	$86.11_{\pm 2.92}$	$\mathbf{100.00_{\pm 0.00}}$
DE-5	$\mathbf{83.31_{\pm 0.14}}$	$\mathbf{61.22_{\pm 0.21}}$	$\mathbf{1.73_{\pm 0.16}}$	$80.65_{\pm 0.19}$	$77.69_{\pm 0.12}$	$83.86_{\pm 0.44}$	$82.33_{\pm 1.34}$	$99.87_{\pm 0.18}$

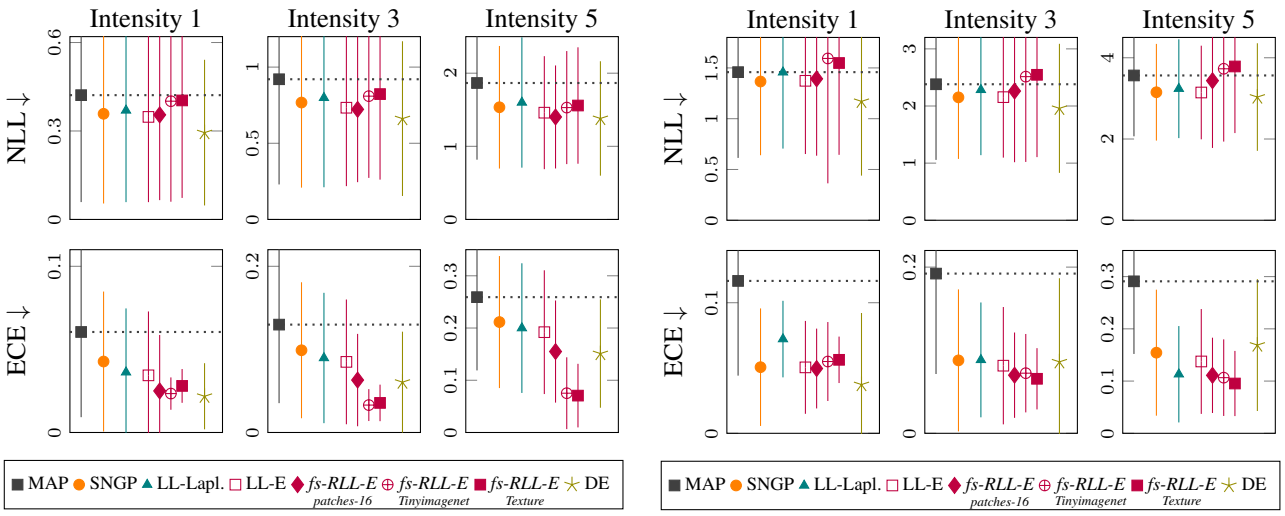


Figure 5. NLL and uncertainty calibration of the different methods on CIFAR10-C (left) and CIFAR100-C (right), for different levels of corruption intensity, averaged over all corruption types. By retraining the last linear layer only, our method fs-RLL-E improves NLL, and achieves similar ECE scores as DEs.

C.5. Covariate shift calibration

We analyze the behavior of our model when presented with corrupted data (CIFAR10-C and CIFAR100-C (Hendrycks & Dietterich, 2019)) for the same base network, Wide-Resnet-28-10. These datasets contain 19 types of corruptions, each with 5 different severity levels. Figure 5 shows the NLL and ECE results averaged over all corruption types. Retraining the last layer without any regularization (LL-E) improves the uncertainty calibration of the single network (MAP) for both datasets. Function space repulsion further improves calibration on CIFAR10-C, outperforming LL-Laplace and SNGP, and achieving competitive uncertainty calibration to DEs. If the final layer of the network is sensitive to features that are not relevant to the target class, corrupted data might activate these irrelevant features, resulting in overconfident and inaccurate predictions. Enforcing predictive diversity on specified repulsion samples can help to down-weight the influence of non-robust features (through the repulsion term) and focus on robust features for prediction, thereby improving the model's calibration and performance on corrupted data. Interestingly, for CIFAR100-C, the benefits of performing inference in function space are not as evident as for CIFAR10-C. Here, random initialization and retraining of the last layer (LL-E) achieves comparable results to the other uncertainty methods at minimal parameter and computational cost.

D. Related work

Scalable Bayesian Neural Networks

Bayesian neural network provide a principled way to quantify uncertainty in neural networks. Unfortunately, the computational cost of training and inference is often prohibitive. Gradient based Monte Carlo methods, such as Hamiltonian Monte Carlo (Neal, 1995), are powerful tools for sampling from complex distributions. However, they are computationally expensive and require careful tuning of hyperparameters. Thus, much research has been invested in finding efficient methods to approximate the posterior distribution and to make BNNs scalable, including variational inference (Blundell et al., 2015), dropout as variational inference (Gal & Ghahramani, 2016), and Laplace approximation (Daxberger et al., 2021). Partially stochastic networks reduce the computational demands of BNNs by treating only a subset of the parameters probabilistically (Daxberger et al., 2021; Sharma et al., 2023). In particular, last-layer approaches have been shown to be effective in reducing overconfidence (Dusenberry et al., 2020; Kristiadi et al., 2020; Harrison et al., 2024). Our multi-headed structure can be interpreted as a partially stochastic network where the last layer is trained using particle optimization in function space.

(Repulsive) Deep Ensembles

Deep ensembles combine the predictions of several deep neural networks, where each network is initialized randomly and trained independently. Originally considered an uncertainty heuristic, DEs have been shown to outperform Bayesian methods in empirical evaluations regarding prediction accuracy, uncertainty calibration, and out-of-distribution detection (Lakshminarayanan et al., 2017; Gustafsson et al., 2020; Ovadia et al., 2019). Subsequently, there has been considerable research on DEs and the conditions under which they be considered a Bayesian method (Wilson, 2020; D'Angelo & Fortuin, 2021; Wild et al., 2023). Repulsive DEs introduce a kernelized repulsion term that prevents ensemble members from collapsing to the same local optimum. They differ in the space in which diversity is enforced: network parameters (Wang et al., 2019; D'Angelo & Fortuin, 2021), feature representations (Yashima et al., 2022), input gradients (Trinh et al., 2023), or function space (Wang et al., 2019; D'Angelo & Fortuin, 2021).

Function-space Inference

A number of inference methods for BNNs consider the shift from inference in the space of network parameters to the function space (Sun et al., 2019; Ma et al., 2019; Burt et al., 2020; Wang et al., 2019; Ma & Hernández-Lobato, 2021; Rudner et al., 2022). This allows to specify meaningful prior distributions over the network parameters. Recent work proposed a tractable variational inference method by linearizing the function mapping of the neural network around a Gaussian distribution (Rudner et al., 2022; 2023). POVI methods approximate the posterior distribution using a set of discrete particles to capture its multimodal structure (Wang et al., 2019; D'Angelo & Fortuin, 2021).

Auxiliary out-of-distribution data

Function space inference methods enforce the function prior on a set of input points, in some work referred to as measurement (Sun et al., 2019; Wang et al., 2019; Ma & Hernández-Lobato, 2021) or context samples (Rudner et al., 2022; 2023). In low-dimensional problems, such samples can be obtained by drawing from a distribution with support over the domain of interest (Sun et al., 2019; Wang et al., 2019; Ma & Hernández-Lobato, 2021). For high-dimensional problems with structured data, such as natural images, samples from an OOD data set have shown improvements (Rudner et al., 2022; 2023). Similar work on OOD detection methods for single networks has used auxiliary OOD datasets to maximize softmax entropy (Hendrycks et al., 2019).

Multi-headed architectures

Various approaches have used multi-headed network architectures to reduce memory requirements by sharing parameters of a base network (Song & Chai, 2018; Sercu et al., 2016; Lee et al., 2015). In reinforcement learning, bootstrapping using a multi-headed network has been employed to improve exploration tasks (Osband et al., 2016). In addition, multi-headed networks were used to perform online distillation of a teacher model (Zhu et al., 2018), and to replicate functional behavior of deep ensembles (Tran et al., 2020). The trade-off between ensembling the whole network and a selection of specific layers has been analyzed in (Valdenegro-Toro, 2023).

Distance based uncertainty methods

Distance-based uncertainty methods consider the epistemic uncertainty of a given test input to be proportional to the distance to the support of the training data. If a test sample is close to the training distribution, the predictions are considered trustworthy. If the distance is large, the model should abstain from making predictions. Computing distances directly in high-dimensional input spaces, however, is often impractical. Thus, most methods depend on well-informed latent representations of the network and estimate epistemic uncertainty by considering feature space densities (Charpentier et al., 2020; Postels et al., 2020; Mukhoti et al., 2023; Winkens et al., 2020) or distances (Liu et al., 2020; van Amersfoort et al., 2020; Tagasovska & Lopez-Paz, 2019). This requires appropriate regularization of the feature space to avoid feature collapse and to ensure that densities and distances are meaningful (van Amersfoort et al., 2021). Common methods to achieve bi-Lipschitz conditions include gradient penalties (Gulrajani et al., 2017; van Amersfoort et al., 2020), and spectral normalization (Miyato et al., 2018; Liu et al., 2020). In extensive experiments, (Postels et al., 2021) demonstrate that relying solely on the feature space density of a model is not sufficient to indicate correctness of a prediction and results in poor calibration under distribution shifts.