Supplemental material:
Bayesian posterior approximation with stochastic ensembles

Oleksandr Balabanov[1], Bernhard Mehlig[2], Hampus Linander[2,3]
[1]Stockholm University [2]University of Gothenburg [3]Chalmers University of Technology
oleksandr.balabanov@fysik.su.se, bernhard.mehlig@physics.gu.se, hampus.linander@gu.se

1. The KL divergence against the prior

Recall that the total loss in Eq. (2) of the main text [10] contains two terms, the KL divergence against the prior $p(\theta)$ and the expected negative log likelihood loss. Here we look at the former term and simplify it.

General case: Mixture of Gaussian distributions. Let us first consider a mixture of Gaussian distributions and then apply the obtained result to deep ensembles and stochastic ensembles. The Gaussian mixture case has been recently addressed in Refs. [1,4]. Here we perform a similar derivation, however, differently to Ref. [4] we do not drop any contributions to the loss and differently to Ref. [1] we derive a simpler upper bound to the repulsive-force correction.

Take a mixture of Gaussian distributions,

$$q(\theta) = \sum_{i=1}^{L} c_i \mathcal{N}(\theta; \mu_i, \sigma_i^2 I_{\text{dim}(\theta)}), \tag{1}$$

where $\mu_i \in \mathbb{R}^{\text{dim}(\theta)}$ is the mean and $\sigma_i \in \mathbb{R}_+$ is the standard deviation corresponding to the multivariate normal distribution $\mathcal{N}(\theta; \mu_i, \sigma_i^2 I_{\text{dim}(\theta)})$, and c_i are some positive constants that sum to 1. Here $i = 1, ..., L$, where L is the number of Gaussians in the mixture.

We are interested in computing the following KL divergence:

$$\begin{aligned} \text{KL}(\, q(\theta) \,||\, p(\theta) \,) &= \int d\theta \, q(\theta) \log \, q(\theta) - \int d\theta \, q(\theta) \log \, p(\theta) \\ &= -H(q(\theta)) - \int d\theta \, q(\theta) \log \, p(\theta), \end{aligned} \tag{2}$$

where $H(q(\theta))$ is the entropy of $q(\theta)$.

General case: The entropy contribution. Let us consider the entropy term

$$\begin{aligned} H(q(\theta)) &= -\sum_{i=1}^{L} c_i \int d\theta \, \mathcal{N}(\theta; \mu_i, \sigma_i^2 I_{\text{dim}(\theta)}) \log \, q(\theta) \\ &= -\sum_{i=1}^{L} c_i \int d\theta \, \mathcal{N}(\theta; 0, I_{\text{dim}(\theta)}) \log \, q(\mu_i + \sigma_i \theta) \\ &= -\sum_{i=1}^{L} c_i \int d\theta \, \mathcal{N}(\theta; 0, I_{\text{dim}(\theta)}) \log \sum_{j=1}^{L} \frac{c_j}{(\sigma_j \sqrt{2\pi})^{\text{dim}(\theta)}} \exp\left[-\frac{||\mu_j - \mu_i - \sigma_i \theta||^2}{2\sigma_j^2} \right] \\ &= -\sum_{i=1}^{L} c_i \int d\theta \, \mathcal{N}(\theta; 0, I_{\text{dim}(\theta)}) \log \sum_{j=1}^{L} \frac{c_j}{(\sigma_j)^{\text{dim}(\theta)}} \exp\left[-\frac{\sigma_i^2}{\sigma_j^2} \frac{||(\mu_j - \mu_i)/\sigma_i - \theta||^2}{2} \right] + \sum_{i=1}^{L} \frac{c_i}{2} \text{dim}(\theta) \log \, 2\pi. \end{aligned} \tag{3}$$

Now, the leading contribution is at $j = i$ and without making any assumptions yet we separate it from the remaining sub-leading term, dubbed RF:

$$\begin{aligned} H(q(\theta)) &= \sum_{i=1}^{L} \frac{c_i}{2} \left[\int d\theta \, \mathcal{N}(\theta; 0, I_{\text{dim}(\theta)}) ||\theta||^2 + \text{dim}(\theta)(\log \, \sigma_i^2 + \log \, 2\pi) \right] - \sum_{i=1}^{L} c_i \log \, c_i - \text{RF} \\ &= \sum_{i=1}^{L} \frac{c_i}{2} \text{dim}(\theta)(1 + \log \, 2\pi + \log \, \sigma_i^2) - \sum_{i=1}^{L} c_i \log \, c_i - \text{RF}, \end{aligned} \tag{4}$$

where

$$\begin{aligned}\mathrm{RF} = &\sum_{i=1}^{L} \int d\theta \, \frac{c_i}{(\sqrt{2\pi})^{\dim(\theta)}} \exp\left(-\frac{||\theta||^2}{2}\right) \\ &\times \log\left(1 + \sum_{j\neq i}^{L} \frac{c_j}{c_i} \frac{(\sigma_i)^{\dim(\theta)}}{(\sigma_j)^{\dim(\theta)}} \exp\left(\frac{||\theta||^2}{2} - \frac{\sigma_i^2}{\sigma_j^2} \frac{||(\mu_j - \mu_i)/\sigma_i - \theta||^2}{2}\right)\right).\end{aligned} \tag{5}$$

Let us assume that $\sigma_i = \sigma$ for all indices i. In this case the RF term reads as

$$\begin{aligned}\mathrm{RF} &= \sum_{i=1}^{L} \int d\theta \, \frac{c_i}{(\sqrt{2\pi})^{\dim(\theta)}} \exp\left(-\frac{||\theta||^2}{2}\right) \log\left(1 + \sum_{j\neq i}^{L} \frac{c_j}{c_i} \exp\left(\frac{||\theta||^2}{2} - \frac{||(\mu_j - \mu_i)/\sigma - \theta||^2}{2}\right)\right) \\ &\leq \sum_{i=1}^{L} \int d\theta \, \frac{c_i}{(\sqrt{2\pi})^{\dim(\theta)}} \exp\left(-\frac{||\theta||^2}{2}\right) \left(\sum_{j\neq i}^{L} \frac{c_j}{c_i} \exp\left(\frac{||\theta||^2}{2} - \frac{||(\mu_j - \mu_i)/\sigma - \theta||^2}{2}\right)\right)^{\frac{1}{2}} \\ &\leq \sum_{i=1}^{L} \sum_{i\neq j} \sqrt{c_i c_j} \int d\theta \, \frac{1}{(\sqrt{2\pi})^{\dim(\theta)}} \left(\exp\left(-\frac{||\theta||^2}{4} - \frac{||(\mu_j - \mu_i)/\sigma - \theta||^2}{4}\right)\right) \\ &= \sum_{i=1}^{L} \sum_{i\neq j} \sqrt{c_i c_j} \exp\left(-\frac{||\mu_j - \mu_i||^2}{8\sigma^2}\right) \int d\theta \, \frac{1}{(\sqrt{2\pi})^{\dim(\theta)}} \exp\left(-\frac{||(\mu_j - \mu_i)/(2\sigma) - \theta||^2}{2}\right) \\ &= \sum_{i=1}^{L} \sum_{j\neq i} \sqrt{c_i c_j} \exp\left(-\frac{||\mu_j - \mu_i||^2}{8\sigma^2}\right).\end{aligned} \tag{6}$$

Thus, the term RF is indeed exponentially small in the limit of small σ and assuming $\mu_i \neq \mu_j$ for all pairs of indices $i \neq j$. Also, note that $\mathrm{RF} \leq \sum_{i=1}^{L} \sum_{i\neq j} \sqrt{c_i c_j} \leq (L-1)$. We can derive a better upper bound than $(L-1)$ by applying the Cauchy–Schwarz inequality for integrals to Eq. (5) with $\sigma_i = \sigma$:

$$\begin{aligned}\mathrm{RF} &= \sum_{i=1}^{L} \int d\theta \, \frac{c_i}{(\sqrt{2\pi})^{\dim(\theta)}} \exp\left(-\frac{||\theta||^2}{2}\right) \log\left(1 + \sum_{j\neq i}^{L} \frac{c_j}{c_i} \exp\left(\frac{||\theta||^2}{2} - \frac{||(\mu_j - \mu_i)/\sigma - \theta||^2}{2}\right)\right) \\ &\leq \sum_{i=1}^{L} \int d\theta \, \frac{c_i}{(\sqrt{2\pi})^{\dim(\theta)}} \exp\left(-\frac{||\theta||^2}{2}\right) \left(\sum_{j\neq i}^{L} \frac{c_j}{c_i} \exp\left(\frac{||\theta||^2}{2} - \frac{||(\mu_j - \mu_i)/\sigma - \theta||^2}{2}\right)\right)^{\frac{1}{2}} \\ &= \sum_{i=1}^{L} \int d\theta \, \frac{1}{(\sqrt{2\pi})^{\dim(\theta)/2}} \exp\left(-\frac{||\theta||^2}{4}\right) \left(\sum_{j\neq i}^{L} \frac{c_i c_j}{(\sqrt{2\pi})^{\dim(\theta)/2}} \exp\left(\frac{||(\mu_j - \mu_i)/\sigma - \theta||^2}{2}\right)\right)^{\frac{1}{2}} \\ &\leq \sum_{i=1}^{L} \left(\int d\theta \, \frac{1}{(\sqrt{2\pi})^{\dim(\theta)}} \exp\left(-\frac{||\theta||^2}{2}\right)\right)^{\frac{1}{2}} \left(\int d\theta \sum_{j\neq i} \frac{c_i c_j}{(\sqrt{2\pi})^{\dim(\theta)}} \left(\exp\left(-\frac{||(\mu_j - \mu_i)/\sigma - \theta||^2}{2}\right)\right)\right)^{\frac{1}{2}} \\ &= \sum_{i=1}^{L} \sqrt{\sum_{j\neq i} c_i c_j} = \sum_{i=1}^{L} \sqrt{c_i(1-c_i)} \leq \sqrt{L-1}.\end{aligned} \tag{7}$$

Thus, the correction RF can be approximated via Eq. (6) that takes form of a repulsive force in the parameter space and it is exponentially small in the limit of small σ.

General case: The prior contribution. Let us consider the normally distributed prior $p(\theta) = \mathcal{N}(\theta; 0, \lambda^{-1} I_{\dim(\theta)})$. Then, the second term in Eq. (2) will simplify to

$$\begin{aligned}\int d\theta\, q(\theta) \log\, p(\theta) &= \sum_{i=1}^{L} c_i \int d\theta\, \mathcal{N}(\theta; \mu_i, \sigma_i^2 I_{\dim(\theta)}) \log\, \mathcal{N}(\theta; 0, \lambda^{-1} I_{\dim(\theta)}) \\ &= -\sum_{i=1}^{L} \frac{c_i}{2} \left[\int d\theta\, \mathcal{N}(\theta; \mu_i, \sigma_i^2 I_{\dim(\theta)})\, \lambda ||\theta||^2 + \dim(\theta) \log\, 2\pi - \dim(\theta) \log \lambda \right] \\ &= -\sum_{i=1}^{L} \frac{c_i}{2} \left[\lambda ||\mu_i||^2 + \dim(\theta) \left(\lambda \sigma_i^2 + \log\, 2\pi - \log \lambda \right) \right].\end{aligned} \tag{8}$$

Deep ensembles. Take a deep ensemble with the corresponding variational inference ansatz

$$q(\theta) = \frac{1}{K} \sum_{k=1}^{K} \mathcal{N}(\theta; \omega_k, \sigma^2 I_{\dim[\theta]}). \tag{9}$$

The KL divergence against the normal prior is then given by

$$\mathrm{KL}(\, q(\theta)\, ||\, p(\theta)\,) = \frac{1}{2} \left[\dim[\theta](\lambda\sigma^2 - \log\, \sigma^2 - 1 - \log \lambda) + \frac{1}{K} \sum_{k=1}^{K} \lambda ||\omega_k||^2 \right] - \log\, K + \mathrm{RF}, \tag{10}$$

with

$$\mathrm{RF} \leq \min \left(\frac{1}{K} \sum_{k=1}^{K} \sum_{k' \neq k} \exp\left(-\frac{||\omega_k - \omega_{k'}||^2}{8\sigma^2} \right), \sqrt{K-1} \right). \tag{11}$$

The repulsive-force contribution RF bounded by Eq. (11) is exponentially small in the limit of small σ. It therefore can be omitted when considering regular deep ensembles with δ-function like members. Note that the repulsive-force upper bound $\sqrt{K-1}$ is larger than the KL loss reduction due to ensembling $\log K$. This is consistent with our expectation that one should not get a reduction in the KL loss by simply rewriting one network as an ensemble of identical members.

Stochastic ensembles. Consider a stochastic ensemble described by the following distribution, see Sec. 5 of the main text [10]:

$$q(\theta) = \frac{1}{K} \sum_{k=1}^{K} \prod_{n=1}^{N} \hat{q}_{\omega_{l,n,k}}(\theta_{l,n}), \tag{12}$$

with

$$\hat{q}_{\omega_{l,n,k}}(\theta_{l,n}) = p^{(1)}_{l+1,\,n}\, \mathcal{N}(\,\theta_{l,n}; \omega^{(1)}_{l,n,k}\,,\, \sigma^2\, I_{\dim[\theta_{l,n}]}\,) + p^{(2)}_{l+1,\,n}\, \mathcal{N}(\,\theta_{l,n}; \omega^{(2)}_{l,n,k}\,,\, \sigma^2\, I_{\dim[\theta_{l,n}]}\,). \tag{13}$$

First, we reduce Eq. (12) to the following expression

$$q(\theta) = \sum_{k=1}^{K} \sum_{\{i_1,\ldots,i_N\}} p^{\{i_1,\ldots,i_N\}}_{l+1}\, \mathcal{N}(\theta; \omega^{\{i_1,\ldots,i_N\}}_{l,k}, \sigma^2 I_{\dim[\theta]}), \tag{14}$$

where

$$\begin{aligned} p^{\{i_1,\ldots,i_N\}}_{l+1} &= \frac{1}{K} \prod_{n=1}^{N} p^{(i_n)}_{l+1,n} \\ \omega^{\{i_1,\ldots,i_N\}}_{l,k} &= [(\omega^{(i_1)}_{l,1,k})^T, (\omega^{(i_2)}_{l,2,k})^T, \ldots, (\omega^{(i_N)}_{l,N,k})^T]^T. \end{aligned} \tag{15}$$

Here we simply collected distinct terms corresponding to different realizations of the parameters and $\{i_1, ..., i_N\}$ labels the indices corresponding to these realizations. $i_n = 1, 2$.

Now, the KL divergence against the normal prior is given by

$$\begin{aligned}
\mathrm{KL}(\, q(\theta)\,||\,p(\theta)\,) &= \frac{1}{2}\left[\dim[\theta]\left(\lambda\sigma^2 - 1 - \log\sigma^2 - \log\lambda\right) + \sum_{k=1}^{K}\sum_{\{i_1,\ldots,i_N\}} p_{l+1}^{\{i_1,\ldots,i_N\}}\lambda||\omega_{l,k}^{\{i_1,\ldots,i_N\}}||^2\right] \\
&+ \sum_{k=1}^{K}\sum_{\{i_1,\ldots,i_N\}} p_{l+1}^{\{i_1,\ldots,i_N\}}\log p_{l+1}^{\{i_1,\ldots,i_N\}} + \mathrm{RF}_2 \\
&= \frac{1}{2}\left[\dim[\theta]\left(\lambda\sigma^2 - 1 - \log\sigma^2 - \log\lambda\right) + \frac{1}{K}\sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{i=1}^{2}\lambda p_{l+1,n}^{(i)}||\omega_{l,n,k}^{(i)}||^2\right] \\
&+ \sum_{n=1}^{N}\sum_{i=1}^{2} p_{l+1,n}^{(i)}\log p_{l+1,n}^{(i)} - \log K + \mathrm{RF}_2,
\end{aligned} \tag{16}$$

with

$$\mathrm{RF}_2 \leq \sum_{k}^{K}\sum_{k'\neq k}\sum_{\{i_1,\ldots,i_N\}}\sum_{\substack{\{j_1,\ldots,j_N\}\\ \neq\{i_1,\ldots,i_N\}}}\sqrt{p_{l+1}^{\{i_1,\ldots,i_N\}}p_{l+1}^{\{j_1,\ldots,j_N\}}}\exp\left(-\frac{||\omega_{l,k}^{\{i_1,\ldots,i_N\}} - \omega_{l,k'}^{\{j_1,\ldots,j_N\}}||^2}{8\sigma^2}\right). \tag{17}$$

The repulsive-force contribution RF_2 is bounded by Eq. (17) that consists of exponentially decaying terms over all possible parameter realizations weighted by the probabilities of obtaining these parameter realizations. This term is exponentially small in the limit of small σ and large N. The conventional dropout (SE1), DropConnect (SE2) and non-parametric dropout (SE3) stochastic ensembles are realized by assuming the infinitesimal (machine-precision) limit of σ [10] and therefore the RF_2 contribution can be dropped in these cases.

2. Metrics

Predictive Entropy. The predictive entropy is defined by

$$H(y^*|x^*, D) = -\sum_c p(y^* = c\,|x^*, \mathcal{D})\log\, p(y^* = c\,|x^*, \mathcal{D}), \tag{18}$$

where x^* is input, $\mathcal{D}$ is the training data, c is the output class index, and $p(c|x^*, \mathcal{D})$ is the posterior output prediction. The output probabilities $p(c|x^*, \mathcal{D})$ are computed by sampling the network parameters θ from the Bayesian posterior $p(\theta|\mathcal{D})$ and then taking the average of softmax outputs.

Mutual Information. The (average) mutual Shannon information contained between network parameters θ and data sample x^* conditioned on the training dataset $\mathcal{D}$ can be conveniently calculated using the following expression [5] in terms of the predictive distribution

$$\mathrm{MI}(\theta, y^*|x^*, D) = H(y^*|x^*, D) - \mathbb{E}_{p(\theta|\mathcal{D})}\left[H(y^*|x^*, \theta)\right]. \tag{19}$$

Thus, $\mathrm{MI}(\theta, y^*|x^*, D)$ is the difference between the posterior entropy given the dataset $\mathcal{D}$ and expected entropy for likelihood $p(y^*|x^*, \theta)$.

Agreement. We use the following formula [9] for computing agreement between two posteriors p_1 and p_2

$$\mathrm{Agr}\,(p_1, p_2) = \frac{1}{n}\sum_{i=1}^{n} I\left[\arg\max_c p_1(y = c\,|x_i) = \arg\max_c p_2(y = c\,|x_i)\right], \tag{20}$$

where $I[\cdot]$ is the indicator function, the sum is over all test data samples x_i, and n is the test dataset size.

Variance. The variance between two posteriors p_1 and p_2 is defined as follows [9]

$$\text{Var}\,(p_1, p_2) = \frac{1}{2n} \sum_{i=1}^{n} \sum_{c} |p_1(y = c\,|x_i) - p_2(y = c\,|x_i)|\,. \tag{21}$$

3. Toy classification problem

Data. The two dimensional training data is produced using the following parametrization: The data contains 2D points $(r\cos(\theta)/20,\ r\sin(\theta)/20)$, where $r = 2\theta + \pi$ for class 1 and $r = 2\theta - \pi$ for class 2. Here we have $\theta = 2\pi\sqrt{\epsilon}$ with uniformly distributed $\epsilon \in [0, 1]$. We also add Gaussian noise of different amplitudes to the three datasets in Fig. 1 of the main text [10]. The chosen noise amplitudes are 0.05, 0.1 and 0.125. For each dataset we produce 2000 data points, 1000 per class. In this way we obtain three training datasets and train each ensemble method on the same data.

The test datasets sample from two domains, in-domain $\mathcal{D}_{\text{in}} \in [-1, 1]^2$ and out-of-domain $\mathcal{D}_{\text{out}} \in [-10, 10]^2$. Each dataset consists of vertices of 100×100 equally spaced grid, in total 10000 points per test dataset.

Model. The model is a feed forward neural network with two hidden layers of 10 neurons, ReLU activations, and softmax output.

HMC. We implement 3 seperate HMC (NUTS) runs corresponding to each of the three toy datasets. Each computation consists of 4 independent HMC chains. We produced 2000 parameter samples per chain after 2000 burn-in steps. The prior variance is chosen to be 1.0.

To verify convergence of our HMC runs we computed the $\hat{R}$ diagnostics [2]. We obtained that 100%, 99%, 90%, and 75% of $\hat{R}$ values corresponding to distinct parameters are smaller than 1.04, 1.02, 1.01, and 1.005 respectively. The $\hat{R}$ values are close to 1.0 indicating good convergence of HMC. We also looked at parallel coordinate and trace plots and did not observe any unwanted patterns. As a last consistency check we looked at the agreement and variance between 1-chain and 4-chain HMC posteriors. We obtained 99.6 , 99.9, 99.9 agreement and 0.006, 0.001, 0.001 variance for the in-domain datasets and 98.5, 99.7, 99.6 agreement and 0.009, 0.005, 0.004 variance for the out-of-domain datasets.

Ensembles. To stay consistent we always followed the same standard training procedure for each ensemble method: We trained for 5000 epochs using Nesterov SGD optimization method with momentum 0.9, the batch size is 100, the starting learning rate is 0.01 that was dropped by a factor of 10 after 2500 and 3750 epochs. We produced 1024 trained networks per ensemble. For the dropout (SE1) and DropConnect (SE2) ensembles we fixed the dropout rate at 0.1 everywhere except the output layer. We also tried to use larger dropout rates but they resulted in worse performance. This is expected for small models.

MultiSWA. In the main text [10] we present results for the MultiSWA variety of MultiSWAG [11]. MultiSWA requires only one inference per network at test time: We load trained networks from the regular ensemble and post train them for additional 2000 epochs with high constant learning rate. We then compute the Stochastic Weight Averaging (SWA) of the parameters to produce the SWA networks. We considered the following constant SWA learning rates: 0.01, 0.005, 0.001, 0.0005 and found the smallest to perform the best. Further reduction of the learning rate is not relevant as it would then be similar to the learning rate of the original models, defeating the purpose of MultiSWA.

4. ResNet-20-FRN evaluated on CIFAR

Data. The training datasets consist of 4096 images randomly selected from CIFAR training datasets. Each network from the same ensemble is trained on the same set of images without any data augmentation but the image selection may differ between the ensembles. We evaluate on the test CIFAR-10 and CIFAR-100 datasets consisting of 10000 images. Robustness to distribution shifts is tested on CIFAR-10-C and CIFAR-100-C. We evaluate on the images from the following 16 corruption categories: fog, zoom blur, speckle noise, glass blur, spatter, shot noise, defocus blur, elastic transform, gaussian blur, frost, saturate, brightness, gaussian noise, contrast, impulse noise, pixelate. For each corruption all 5 intensity levels are considered.

Model. The model is ResNet-20-FRN that is a residual network of depth 20 with batch normalization layers replaced with filter response normalization (FRN). We note that we had to explicitly adjust the paddings of the convolutional layers with

	Entropy 1e-3	MI 1e-3	Agr 1e-2	Var 1e-2	Entropy 1e-3	MI 1e-3	Agr 1e-2	Var 1e-2	Entropy 1e-3	MI 1e-3	Agr 1e-2	Var 1e-2
	Toy-a (in-domain / out-of-domain)				Toy-b (in-domain / out-of-domain)				Toy-c (in-domain / out-of-domain)			
NP Dropout (SE3)	0.46 1.14	0.38 1.13	98.4 95.6	2.59 7.00	0.25 0.73	0.15 0.71	98.9 95.8	1.41 4.51	0.22 0.76	0.11 0.73	99.0 96.7	1.20 3.86

Table 1. Quantitative comparison of predictions obtained from HMC and non-parametric dropout ensemble SE3 with 8 test-time inferences per model. The tests are done using data from $\mathcal{D}_{\text{in}} = [-1,\ 1]^2$ (in-domain, top rows) and $\mathcal{D}_{\text{out}} = [-10,\ 10]^2$ (out-of-domain, bottom rows). We considered all three different toy datasets. The considered metrics are agreement, variance, mean absolute difference of entropy and mutual information estimates computed in respect to the full HMC runs. All variances are orders of magnitude smaller than quoted results given the large ensemble sizes.

stride 2 in our PyTorch implementation to create an exact replica of ResNet-20-FRN architecture implemented in Ref. [9] using Jax. This is because the padding for these layers contains a certain implementation mismatch in PyTorch and Jax.

Hamiltonian Monte Carlo. The HMC chains were loaded from the publicly available resource [8, 9]. For CIFAR-10 we loaded 3 HMC chains of 291 checkpoints each. The first 50 of 291 checkpoints are used as burn-in as explained in Ref. [9] so in total we used 723 parameter sets. Similarly, for CIFAR-100 we loaded 3 HMC chains of 200 checkpoints each (v2) and neglected first 50 burn-in checkpoints. In total we evaluated using 450 parameter realizations in this case. For both datasets the Gaussian prior has variance 0.2. For more details on the HMC computations see Ref. [9].

Ensembles. The training was done in the same way for each method. We followed a training procedure similar to Refs. [3, 6, 7]: Each ensemble member was trained for 600 epochs with learning rate 0.1 that was dropped by a factor 10 after 200 and 400 epochs. We used Nesterov SGD optimizer with momentum 0.9. The batch size is 128. No early stopping was used. L2 regularization was adjusted to match the Gaussian prior used in the HMC runs.

For the Monte Carlo dropout (SE1) and DropConnect (SE2) stochastic ensembles we applied a non-zero drop rate only to the convolutional layers. The output and FRN layers were left unmodified because the considered parametric stochastic methods SE1 and SE2 cannot be straightforwardly applied to these layers. The dropout operation was implemented by randomly dropping individual neurons rather than neuron layers. We implemented SE1 and SE2 ensembles with drop rates ranging from 0.1 to 0.5 and selected the drop rates producing the most accurate posteriors. For the ensembles trained on CIFAR-10 the drop rate was tuned to 0.3 for SE1 and 0.2 for SE2. The network was altered in an analogous way for CIFAR-100 but with drop rates 0.2 for both SE1 and SE2 ensembles. The non-parametric dropout ensemble SE3 was implemented by applying the non-parametric dropout operation to every layer in the network, i.e. to each convolutional, linear and FRN layer.

MultiSWA. For CIFAR classification we also implement the MultiSWA protocol and use its results as one of the baselines. MultiSWA is implemented by post training the networks from the regular ensemble with some high learning rate. In our tests we considered 0.05, 0.01, 0.005 constant learning rate schedulers and post trained for 300 epochs while saving the average. For both CIFAR-10 and CIFAR-100 the learning rate 0.005 was found to perform the best.

Out-Of-Distribution Detection. To test the out-of-distribution detection (ODD) we train the ensembles on CIFAR-10 or CIFAR-100 but test on a combined test dataset of CIFAR-10 and CIFAR-100. The ensembles are then asked to detect the out-of-training data: The test images are sorted by the highest output probability, expecting the in-domain data to end up at the top of the sorted list. The performance is quantified by calculating AUC-ROC as in Ref. [9].

5. Multiple test-time inferences per ensemble member

Toy model. In the main text [10] we always use one test-time inference per ensemble member. Here we complement these results by implementing more test-time inferences. We choose to do 8 test-time inferences per member. The corresponding data is provided in Table 1. We obtained exactly the same metric values (up to the assumed precision), indicating good convergence of the posterior approximation for the chosen ensemble size 1024.

	Acc	Loss	ECE	Agr 1e-2	Var 1e-2	ODD	Acc	Loss	ECE	Agr 1e-2	Var 1e-2	ODD
	CIFAR-10						CIFAR-100					
Dropout (SE1)	90.74 ± 0.03	0.299 ± 0.002	0.057 ± 0.001	94.2 ± 0.1	7.8 ± 0.1	85.6 ±0.1	68.78 ± 0.32	1.156 ± 0.001	0.128 ± 0.004	77.6 ± 0.2	21.5 ± 0.1	73.7 ±0.1
	CIFAR-10-C (mean over corruptions)						CIFAR-100-C (mean over corruptions)					
Dropout (SE1)	76.45 ± 0.07	0.704 ± 0.002	0.065 ± 0.001	83.3 ± 0.1	16.0 ± 0.1		48.56 ± 0.05	2.084 ± 0.003	0.091 ± 0.001	61.5 ± 0.1	31.2 ± 0.1	

Table 2. Prediction accuracy (acc), test log-likelihood loss (loss), expected calibration error (ECE), agreement (agr), variance (var) and out-of-domain detection (ODD) for stochastic ensembles based on Monte Carlo dropout (SE1) trained on the CIFAR datasets and evaluated on the plain and corrupted CIFAR test datasets.

CIFAR. In the main text [10] only one inference at test time per ensemble member was considered. Here we present the results obtained using 8 test-time inferences per member and list them in Table 2. The obtained values are closer to the corresponding HMC data presented in Tables 2 and 3 of the main text [10] but the improvement is minor given that we use 8 times larger number of samples at test time.

References

[1] Takashi Furuya, Hiroyuki Kusumoto, Koichi Taniguchi, Naoya Kanno, and Kazuma Suetake. Variational inference with gaussian mixture by entropy approximation, 2022. 2

[2] Andrew Gelman and Donald B. Rubin. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7(4):457 – 472, 1992. 6

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 7

[4] Lara Hoffmann and Clemens Elster. Deep Ensembles from a Bayesian Perspective. *arXiv e-prints*, page arXiv:2105.13283, May 2021. 2

[5] Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *CoRR*, abs/1112.5745, 2011. 5

[6] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018. 7

[7] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth, 2016. 7

[8] Pavel Izmailov. https://izmailovpavel.github.io/neurips_bdl_competition/, 2021. 7

[9] Pavel Izmailov, Sharad Vikram, Matthew D. Hoffman, and Andrew Gordon Wilson. What Are Bayesian Neural Network Posteriors Really Like? *arXiv e-prints*, page arXiv:2104.14421, Apr. 2021. 5, 6, 7

[10] Main text. 2, 4, 5, 6, 7, 8

[11] Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708, 2020. 6