COLD POSTERIORS THROUGH PAC-BAYES

Anonymous authors
Paper under double-blind review

A PROOFS MAIN RESULTS

A.1 PROOF OF PROPOSITION 1

Recall that we model our predictor as $f_{\text{lin}}(\boldsymbol{x}; \mathbf{w}) = f(\boldsymbol{x}; \mathbf{w}_{\hat{\rho}}) - \nabla_{\mathbf{w}} f(\boldsymbol{x}; \mathbf{w}_{\hat{\rho}})^\top (\mathbf{w} - \mathbf{w}_{\hat{\rho}})$. Then for the choice of a Gaussian likelihood, given a training signal $\boldsymbol{x}$, a training label y and weights $\mathbf{w}$, the negative log-likelihood loss takes the form $\ell_{\text{nll}}(\mathbf{w}, \boldsymbol{x}, y) = \frac{1}{2}\ln(2\pi\sigma^2) + \frac{1}{2\sigma^2}(y - f(\boldsymbol{x}; \mathbf{w}_{\hat{\rho}}) - \nabla_{\mathbf{w}} f(\boldsymbol{x}; \mathbf{w}_{\hat{\rho}})^\top (\mathbf{w} - \mathbf{w}_{\hat{\rho}}))^2$. We also define $\hat{\mathcal{L}}^{\ell}_{X,Y}(f) = (1/n)\sum_i \ell(f, \boldsymbol{x}_i, y_i)$. Our derivations closely follow the approach of Germain et al. (2016) p.11, section A.4.

Given the above definitions and modelling choices we develop the empirical risk term

$$
\begin{aligned}
2n\sigma^2 \mathbf{E}_{\mathbf{w}\sim\hat{\rho}} \hat{\mathcal{L}}^{\ell_{\text{nll}}}_{X,Y}(\mathbf{w}) - n\sigma^2 \ln(2\pi\sigma^2) &= \mathbf{E}_{\mathbf{w}\sim\hat{\rho}} \sum_{i=1}^{n} (y_i - f(\boldsymbol{x}_i; \mathbf{w}_{\hat{\rho}}) - \nabla_{\mathbf{w}} f(\boldsymbol{x}_i; \mathbf{w}_{\hat{\rho}})^\top (\mathbf{w} - \mathbf{w}_{\hat{\rho}}))^2 \\
&= \mathbf{E}_{\mathbf{w}\sim\hat{\rho}} \| \boldsymbol{y} - f(\mathbf{X}; \mathbf{w}_{\hat{\rho}}) - \nabla_{\mathbf{w}} f(\mathbf{X}; \mathbf{w}_{\hat{\rho}})^\top (\mathbf{w} - \mathbf{w}_{\hat{\rho}}) \|_2^2 \\
&= \mathbf{E}_{\mathbf{w}\sim\hat{\rho}} [\| \boldsymbol{y} - f(\mathbf{X}; \mathbf{w}_{\hat{\rho}}) \|_2^2 - 2(\boldsymbol{y} - f(\mathbf{X}; \mathbf{w}_{\hat{\rho}}))^\top \nabla_{\mathbf{w}} f(\mathbf{X}; \mathbf{w}_{\hat{\rho}})^\top (\mathbf{w} - \mathbf{w}_{\hat{\rho}}) \\
&\quad + (\mathbf{w} - \mathbf{w}_{\hat{\rho}})^\top \nabla_{\mathbf{w}} f(\mathbf{X}; \mathbf{w}_{\hat{\rho}}) \nabla_{\mathbf{w}} f(\mathbf{X}; \mathbf{w}_{\hat{\rho}})^\top (\mathbf{w} - \mathbf{w}_{\hat{\rho}})] \\
&= \mathbf{E}_{\mathbf{w}\sim\hat{\rho}} [\| \boldsymbol{y} - f(\mathbf{X}; \mathbf{w}_{\hat{\rho}}) \|_2^2 - 2(\boldsymbol{y} - f(\mathbf{X}; \mathbf{w}_{\hat{\rho}}))^\top \nabla_{\mathbf{w}} f(\mathbf{X}; \mathbf{w}_{\hat{\rho}})^\top (\mathbf{w} - \mathbf{w}_{\hat{\rho}}) \\
&\quad + (\mathbf{w} - \mathbf{w}_{\hat{\rho}})^\top \left[\textstyle\sum_i \nabla_{\mathbf{w}} f(\boldsymbol{x}_i; \mathbf{w}_{\hat{\rho}}) \nabla_{\mathbf{w}} f(\boldsymbol{x}_i; \mathbf{w}_{\hat{\rho}})^\top\right] (\mathbf{w} - \mathbf{w}_{\hat{\rho}})] \\
&= \mathbf{E}_{\mathbf{w}\sim\hat{\rho}} [\| \boldsymbol{y} - f(\mathbf{X}; \mathbf{w}_{\hat{\rho}}) \|_2^2] - 2(\boldsymbol{y} - f(\mathbf{X}; \mathbf{w}_{\hat{\rho}}))^\top \nabla_{\mathbf{w}} f(\mathbf{X}; \mathbf{w}_{\hat{\rho}})^\top \cancel{\mathbf{E}_{\mathbf{w}\sim\hat{\rho}}[\mathbf{w} - \mathbf{w}_{\hat{\rho}}]} \\
&\quad + \mathbf{E}_{\mathbf{w}\sim\hat{\rho}} \left[(\mathbf{w} - \mathbf{w}_{\hat{\rho}})^\top \left[\textstyle\sum_i \nabla_{\mathbf{w}} f(\boldsymbol{x}_i; \mathbf{w}_{\hat{\rho}}) \nabla_{\mathbf{w}} f(\boldsymbol{x}_i; \mathbf{w}_{\hat{\rho}})^\top\right] (\mathbf{w} - \mathbf{w}_{\hat{\rho}}) \right] \\
&= \| \boldsymbol{y} - f(\mathbf{X}; \mathbf{w}_{\hat{\rho}}) \|_2^2 + \sigma_{\hat{\rho}}^2 \left[\textstyle\sum_i \sum_j (\nabla_{\mathbf{w}} f(\boldsymbol{x}_i; \mathbf{w}_{\hat{\rho}})_j)^2 \right] \\
&= \| \boldsymbol{y} - f(\mathbf{X}; \mathbf{w}_{\hat{\rho}}) \|_2^2 + \sigma_{\hat{\rho}}^2 h.
\end{aligned}
$$

In the penultimate line, we have used the fact that a real number is the trace of itself as well as the cyclic property of the trace. The second summation ($\sum_j$ over the parameters of the model) results from the fact that $\hat{\rho} = \mathcal{N}(\mathbf{w}_{\hat{\rho}}, \sigma_{\hat{\rho}}^2 \mathbf{I})$ is isotropic with a common scaling factor $\sigma_{\hat{\rho}}^2$. The term in blue is exactly the Gauss–Newton approximation to the Hessian of the full neural network, for the squared loss function (Kunstner et al., 2019; Immer et al., 2021), and in the last line we set $h = \left[\sum_i \sum_j (\nabla_{\mathbf{w}} f(\boldsymbol{x}_i; \mathbf{w}_{\hat{\rho}})_j)^2 \right]$. Since h is a sum of positive numbers, taking into account that the blue term is the Gauss–Newton approximation to the Hessian and if we assume that the Gauss–Newton approximation is diagonal, then h is a measure of the curvature at minimum $\mathbf{w}_{\hat{\rho}}$ of the loss landscape. We finally get

$$
\mathbf{E}_{\mathbf{w}\sim\hat{\rho}} \hat{\mathcal{L}}^{\ell_{\text{nll}}}_{X,Y}(\mathbf{w}) = \frac{\| \boldsymbol{y} - f(\mathbf{X}; \mathbf{w}_{\hat{\rho}}) \|_2^2}{2n\sigma^2} + \frac{\sigma_{\hat{\rho}}^2 h}{2n\sigma^2} + \frac{1}{2}\ln(2\pi\sigma^2).
$$

We continue with the KL term which is known to have the following analytical expression for Gaussian prior and posterior distributions

$$
\mathrm{KL}(\mathcal{N}(\mathbf{w}_{\hat{\rho}}, \sigma_{\hat{\rho}}^2 \mathbf{I}) \| \mathcal{N}(\mathbf{w}_{\pi}, \sigma_{\pi}^2 \mathbf{I})) = \frac{1}{2}\left(d\frac{\sigma_{\hat{\rho}}^2}{\sigma_{\pi}^2} + \frac{1}{\sigma_{\pi}^2}\| \mathbf{w}_{\hat{\rho}} - \mathbf{w}_{\pi} \|^2 - d - d\ln\frac{\sigma_{\hat{\rho}}^2}{\sigma_{\pi}^2} \right).
$$

We finally develop the moment term. Using an intermediate variable $\lambda_n = \frac{\lambda n}{2}$ to simplify the calculations, we get

$$\begin{aligned}
\Psi_{\ell,\pi,\mathcal{D}}(\lambda, n) &= \ln \mathbf{E}_{f\sim\pi} \mathbf{E}_{(X',Y')\sim\mathcal{D}^n} \exp\left[\lambda n \left(\mathcal{L}_{\mathcal{D}}^{\ell_{\text{nll}}}(f) - \hat{\mathcal{L}}_{X',Y'}^{\ell_{\text{nll}}}(f)\right)\right] \\
&= \ln \mathbf{E}_{f\sim\pi} \mathbf{E}_{(X',Y')\sim\mathcal{D}^n} \exp\left[\lambda_n \left(\mathbf{E}_{(\boldsymbol{x},y)}\left[\ln(2\pi) + (y - f_{\text{lin}}(\boldsymbol{x};\mathbf{w})^2\right]\right.\right. \\
&\qquad \left.\left. -\ \ln(2\pi) - (1/n)\textstyle\sum_i (y_i - f_{\text{lin}}(\boldsymbol{x}_i;\mathbf{w})^2)\right)\right] \\
&= \ln \mathbf{E}_{f\sim\pi} \mathbf{E}_{(X',Y')\sim\mathcal{D}^n} \exp\left[\lambda_n \left(\mathbf{E}_{(\boldsymbol{x},y)}\left[(y - f_{\text{lin}}(\boldsymbol{x};\mathbf{w})^2\right] - (1/n)\textstyle\sum_i (y_i - f_{\text{lin}}(\boldsymbol{x}_i;\mathbf{w})^2)\right)\right] \\
&\leq \ln \mathbf{E}_{\mathbf{w}\sim\pi} \exp\left[\lambda_n \mathbf{E}_{(\boldsymbol{x},y)} \left(y - f_{\text{lin}}(\boldsymbol{x};\mathbf{w})\right)^2\right] \\
&= \ln \mathbf{E}_{\mathbf{w}\sim\pi} \exp[\lambda_n \mathbf{E}_{(\boldsymbol{x},y)} (f(\boldsymbol{x};\mathbf{w}_{\hat{\rho}}) + \nabla_{\mathbf{w}} f(\boldsymbol{x};\mathbf{w}_{\hat{\rho}})^\top (\mathbf{w}_* - \mathbf{w}_{\hat{\rho}}) + \epsilon \\
&\qquad - (f(\boldsymbol{x};\mathbf{w}_{\hat{\rho}}) + \nabla_{\mathbf{w}} f(\boldsymbol{x};\mathbf{w}_{\hat{\rho}})^\top (\mathbf{w} - \mathbf{w}_{\hat{\rho}})))^2] \\
&= \ln \mathbf{E}_{\mathbf{w}\sim\pi} \exp[\lambda_n \mathbf{E}_{(\boldsymbol{x},y)} (\nabla_{\mathbf{w}} f(\boldsymbol{x};\mathbf{w}_{\hat{\rho}})^\top (\mathbf{w}_* - \mathbf{w}) + \epsilon)^2] \\
&= \ln \mathbf{E}_{\mathbf{w}\sim\pi} \exp[\lambda_n \mathbf{E}_{\boldsymbol{x}} [(\nabla_{\mathbf{w}} f(\boldsymbol{x};\mathbf{w}_{\hat{\rho}})^\top (\mathbf{w}_* - \mathbf{w}))^2] + \lambda_n \sigma_\epsilon^2].
\end{aligned}$$

Inequality in line 4 is because the exponential function is less than 1 on the negative half line. In the fifth line we use our modelling choice $y = f(\boldsymbol{x};\mathbf{w}_{\hat{\rho}}) + \nabla_{\mathbf{w}} f(\boldsymbol{x};\mathbf{w}_{\hat{\rho}})^\top (\mathbf{w}_* - \mathbf{w}_{\hat{\rho}}) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. To obtain the final line we note that the gradient of the *neural network output* with respect to $\mathbf{w}$, that is $\nabla_{\mathbf{w}} f(\boldsymbol{x};\mathbf{w}_{\hat{\rho}})$, does *not* depend on the label y. We get the last line by applying the square and taking the expectation, given that the noise ϵ is centered.

We now take into account the Gaussian mixture modelling for the gradients per data sample, $\nabla_{\mathbf{w}} f(\boldsymbol{x};\mathbf{w}_{\hat{\rho}}) \sim \sum_{j=1}^k \phi_j \mathcal{N}(\boldsymbol{\mu}_j, \sigma_{\boldsymbol{x}j}^2 \mathbf{I})$. We get

$$\begin{aligned}
&\mathbf{E}_{\boldsymbol{x}}[(\nabla_{\mathbf{w}} f(\boldsymbol{x};\mathbf{w}_{\hat{\rho}})^\top (\mathbf{w}_* - \mathbf{w}))^2] = \mathbf{E}_{\boldsymbol{x}}[(\textstyle\sum_i \nabla_{\mathbf{w}} f(\boldsymbol{x};\mathbf{w}_{\hat{\rho}})_i (\mathbf{w}_* - \mathbf{w})_i)^2] \\
&\quad = \mathbf{E}_{\boldsymbol{x}}\left[(\textstyle\sum_i \nabla_{\mathbf{w}} f(\boldsymbol{x};\mathbf{w}_{\hat{\rho}})_i^2 (\mathbf{w}_* - \mathbf{w})_i^2 + 2\sum_{i,j} \nabla_{\mathbf{w}} f(\boldsymbol{x};\mathbf{w}_{\hat{\rho}})_i \nabla_{\mathbf{w}} f(\boldsymbol{x};\mathbf{w}_{\hat{\rho}})_j (\mathbf{w}_* - \mathbf{w})_i (\mathbf{w}_* - \mathbf{w})_j)\right] \\
&\quad = \textstyle\sum_i \mathbf{E}_{\boldsymbol{x}}[\nabla_{\mathbf{w}} f(\boldsymbol{x};\mathbf{w}_{\hat{\rho}})_i^2] (\mathbf{w}_* - \mathbf{w})_i^2 = \sum_i \sum_{j=1}^k (\phi_j \sigma_{\boldsymbol{x}j}^2)(\mathbf{w}_* - \mathbf{w})_i^2 = \sigma_{\boldsymbol{x}}^2 \|\mathbf{w}_* - \mathbf{w}\|_2^2.
\end{aligned}$$

The red term cancels out because we assumed that each weight is independent from the others. Next we use the Gaussian mixture modelling to get $\mathbf{E}_{\boldsymbol{x}}[\nabla_{\mathbf{w}} f(\boldsymbol{x};\mathbf{w}_{\hat{\rho}})_i^2] = \sum_{j=1}^k (\phi_j \sigma_{\boldsymbol{x}j}^2)$, and we finally set $\sigma_{\boldsymbol{x}}^2 = \sum_{j=1}^k (\phi_j \sigma_{\boldsymbol{x}j}^2)$, as each component of the mixture is isotropic, thus the second moment of all weights is the same. By completing the square above, one obtains the Gaussian expectation of this squared norm and forms the moment term as follows

$$\begin{aligned}
\Psi_{\ell,\pi,\mathcal{D}}(\lambda, n) &= \ln \mathbf{E}_{\mathbf{w}\sim\pi} \exp\left[\lambda_n \sigma_{\boldsymbol{x}}^2 \|\mathbf{w}_* - \mathbf{w}\|_2^2 + \lambda_n \sigma_\epsilon^2\right] \\
&= \ln\left(\frac{1}{(1 - 2\lambda_n \sigma_{\boldsymbol{x}}^2 \sigma_\pi^2)^{\frac{d}{2}}} \exp\left[\frac{\lambda_n \sigma_{\boldsymbol{x}}^2 \|\mathbf{w}_* - \mathbf{w}_\pi\|_2^2}{1 - 2\lambda_n \sigma_{\boldsymbol{x}}^2 \sigma_\pi^2} + \lambda_n \sigma_\epsilon^2\right]\right) \\
&= -\frac{d}{2} \ln(1 - 2\lambda_n \sigma_{\boldsymbol{x}}^2 \sigma_\pi^2) + \frac{\lambda_n \sigma_{\boldsymbol{x}}^2 \|\mathbf{w}_* - \mathbf{w}_\pi\|_2^2}{1 - 2\lambda_n \sigma_{\boldsymbol{x}}^2 \sigma_\pi^2} + \lambda_n \sigma_\epsilon^2 \\
&\leq \frac{\lambda_n \sigma_{\boldsymbol{x}}^2 \sigma_\pi^2 d}{1 - 2\lambda_n \sigma_{\boldsymbol{x}}^2 \sigma_\pi^2} + \frac{\lambda_n \sigma_{\boldsymbol{x}}^2 \|\mathbf{w}_* - \mathbf{w}_\pi\|_2^2}{1 - 2\lambda_n \sigma_{\boldsymbol{x}}^2 \sigma_\pi^2} + \lambda_n \sigma_\epsilon^2 \\
&= \frac{\lambda_n \sigma_{\boldsymbol{x}}^2 (\sigma_\pi^2 d + \|\mathbf{w}_* - \mathbf{w}_\pi\|_2^2)}{1 - 2\lambda_n \sigma_{\boldsymbol{x}}^2 \sigma_\pi^2} + \lambda_n \sigma_\epsilon^2,
\end{aligned}$$

which assumes $1 - 2\lambda_n \sigma_{\boldsymbol{x}}^2 \sigma_\pi^2 > 0$. The second line above is obtained by using the moment generating function of noncentral χ^2 variables, while the inequality comes from $\ln(u) < u - 1$ for $u > 1$. Setting back $\frac{\lambda n}{2}$ in place of λ_n, we get

$$\frac{1}{\lambda n} \Psi_{\ell,\pi,\mathcal{D}}(\lambda, n) \leq \frac{\sigma_{\boldsymbol{x}}^2 (\sigma_\pi^2 d + \|\mathbf{w}_* - \mathbf{w}_\pi\|_2^2)}{2 - 2\lambda n 2\sigma_{\boldsymbol{x}}^2 \sigma_\pi^2} + \frac{\sigma_\epsilon^2}{2}.$$

We are now ready to minimize the following objective, where the moment term is absent since it does not depend on $\sigma_{\hat{\rho}}^2$

$$\min_{\sigma_{\hat{\rho}}^2} \mathbf{E}_{\mathbf{w}\sim\hat{\rho}} \hat{\mathcal{L}}_{X,Y}^{\ell_{\text{nll}}}(\mathbf{w}) + \frac{1}{\lambda n}\left[\text{KL}(\mathcal{N}(\mathbf{w}_{\hat{\rho}}, \sigma_{\hat{\rho}}^2\mathbf{I}) \| \mathcal{N}(\mathbf{w}_{\pi}, \sigma_{\pi}^2\mathbf{I})) + \ln\frac{1}{\delta}\right]$$

The derivative of the objective function w.r.t. $\sigma_{\hat{\rho}}^2$ simply writes

$$\begin{aligned}
&\frac{\partial}{\partial\sigma_{\hat{\rho}}^2}\left(\frac{\|\boldsymbol{y} - f(\mathbf{X}; \mathbf{w}_{\hat{\rho}})\|_2^2}{2n\sigma^2} + \frac{\sigma_{\hat{\rho}}^2 h}{2n\sigma^2} + \frac{1}{2}\ln(2\pi\sigma^2)\right. \\
&\qquad \left. + \frac{1}{\lambda n}\left[\frac{1}{2}\left(\frac{1}{\sigma_{\pi}^2} d\sigma_{\hat{\rho}}^2 + \frac{1}{\sigma_{\pi}^2}\|\mathbf{w}_{\hat{\rho}} - \mathbf{w}_{\pi}\|_2^2 - d - d\ln\sigma_{\hat{\rho}}^2 + d\ln\sigma_{\pi}^2\right) + \ln\frac{1}{\delta}\right]\right) \\
&\qquad = \frac{h}{2n\sigma^2} + \frac{1}{2\lambda n}\left(\frac{d}{\sigma_{\pi}^2} - \frac{d}{\sigma_{\hat{\rho}}^2}\right).
\end{aligned}$$

Now setting the above to zero we get the typical prior-to-posterior update for a Gaussian precision term

$$\frac{1}{\sigma_{\hat{\rho}}^2} = \frac{\lambda h}{d\sigma^2} + \frac{1}{\sigma_{\pi}^2}.$$

The proposition is proven by replacing the terms in the bound from Theorem 1 with the results derived above.

A.2 COROLLARY 1

Corollary 1. *For* $\sigma^2 = n = d = h = \sigma_{\pi}^2 = \sigma_{\boldsymbol{x}}^2 = \|\mathbf{w}_*\|_2^2 = \|\mathbf{w}_{\hat{\rho}} - \mathbf{w}_{\pi}\| = \sigma_{\epsilon}^2 = 1$, $\|\boldsymbol{y} - f(\mathbf{X}; \mathbf{w}_{\hat{\rho}})\|_2^2 = 0$, *and ignoring additive constants, the dependence of the Proposition 1 bound* $\mathcal{B}_{\text{approximate}}$ *on the temperature parameter* $\lambda \in (0, 1/2)$ *is as follows, with probability at least* $1 - \delta$

$$\mathbf{E}_{\mathbf{w}\sim\hat{\rho}} \mathcal{L}_{\mathcal{D}}^{\ell_{\text{nll}}}(\mathbf{w}) \leq \underbrace{\frac{1}{2}\frac{1}{\lambda+1}}_{\text{Empirical Risk}} + \underbrace{\frac{2}{1-2\lambda}}_{\text{Moment}} + \underbrace{\frac{1}{\lambda}\left[\frac{1}{2}\left(\frac{1}{\lambda+1} + \ln(\lambda+1)\right) + \ln\frac{1}{\delta}\right]}_{\text{KL}}. \tag{1}$$

Proof. The result is directly obtained by a simple inspection of Proposition 1. ☐

From this we see that as we increase λ the Empirical Risk term should decrease, while the Moment term should increase. In this particular case the KL also decreases. We see in later experiments that this intuition is roughly correct, although on different scales for each term.

B EXPERIMENTS

Includes the full set of experimental results along with experiments on regression datasets.

B.1 EXPERIMENTAL SETUP

We run our experiments on GPUs of the type NVIDIA GeForce RTX2080ti, on our local cluster. The total computation time was approximately 125 GPU hours. In the following list we include the libraries and datasets that we used together with their corresponding licences

- Laplace-Redux Package (Daxberger et al., 2021a): MIT License
- Netcal package (Küppers et al., 2021): Apache Software License
- Pytorch package (Paszke et al., 2019): Modified BSD Licence
- Abalone, Diamonds datasets (Dua & Graff, 2017): -
- KC_House datasets (harlfoxem, 2014): CC0, Public Domain

- MNIST-10 dataset (Deng, 2012): MIT Licence
- CIFAR-10 dataset (Krizhevsky & Hinton, 2009): MIT Licence
- CIFAR-100 dataset (Krizhevsky & Hinton, 2009): MIT Licence
- SVHN dataset (Netzer et al., 2011): -
- FashionMnist dataset (Xiao et al., 2017): MIT Licence

B.2 DATASET SPLITS

In all regression experiments we will split the dataset into 4 sets: Z_{train} the training set, Z_{test} the testing set, $Z_{\text{validation}}$ the validation set, Z_{true} a large sample set that is used to approximate the complete distributions. We detail the use of each split in the following sections; refer to Table 1 for the specifics of splits for each dataset.

	Z_{train}	Z_{test}	$Z_{\text{validation}}$	Z_{true}
Abalone	751	835	84	2000
KC_House	3923	4323	400	10406
Diamonds	9788	10788	1000	25970

Table 1: In this table we detail the number of samples that we add to each set of our split, for each dataset. We aim to have a sufficiently high number of samples for the Z_{true}. $Z_{\text{validation}}$ is chosen to be approximately 10% of the Z_{train} (note that $Z_{\text{validation}}$ contains new samples). For the regression datasets our training set is approximately the same size as the testing set which is not a common setup in classification. However, our aim is not to obtain the best training and testing error but to investigate the behaviour of our models for varying λ.

For the classification datasets CIFAR-10, CIFAR-100, SVHN, FMNIST we used the standard test and train splits. We use 10% of the data for the validation set.

B.3 MODELS

For our regression datasets we use a fully connected network with two hidden layers with 100 neurons each and the ReLU non-linearity. We train our networks to minimize the Mean Square Error (MSE) loss. We evaluate the NLL with a Gaussian likelihood with $\sigma = 1$.

For the classification datasets CIFAR-10, CIFAR-100 and SVHN we use the WideResNet22 (Zagoruyko & Komodakis, 2016) architecture. Because the Laplace approximation does not interact well Antorán et al. (2022) with BatchNorm (Ioffe & Szegedy, 2015) we instead use Fixup Initialization Zhang et al. (2019). We train our networks using the softmax activation and the cross-entropy loss. We use the SGD optimizer with learning rate $\eta = 0.1$, weight decay 5e-4, and momentum 0.9 and 300 epochs. We furthermore divide the initial learning rate by 10, at the point of 50%, 75% and 87% of the epochs. We also use dropout with 0.4 after all the Resnet blocks. We evaluate the NLL using the cross-entropy loss.

For the classification dataset FMNIST we use a Convolutional Network with 3 nonlinear convolutional layers followed by 2 non-linear fully connected layers. We use the SGD optimizer with learning rate $\eta = 0.001$, weight decay 5e-4, and momentum 0.9 and 10 epochs. We evaluate the NLL using the cross-entropy loss.

We *do not* use data augmentation in any experiment. This partially explains the problems with the CIFAR-100 dataset. In particular, in preliminary experiments (which we include further in the Appendix) both the CIFAR-10 and the CIFAR-100 dataset improve significantly in accuracy with data augmentation (random flips and random crops) and the matrix inversion in the CIFAR-100 KFAC case is better posed and results in significantly improved accuracy 70% over the non augmented counterpart.

	Average MAP Test Error
CIFAR-10	10.4%
CIFAR-100	40.6%
SVHN	4.2%
FMNIST	8.8%

Table 2: In this table we plot the average test 0-1 Loss of the MAP estimates of the different networks and datasets.

B.4 EVALUATION OF BOUNDS

B.4.1 ALL APPROXIMATE BOUND EVALUATION $\mathcal{B}_{\text{approximate}}$

We need to evaluate the following bound

$$\begin{aligned}
&\mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\mathcal{L}_{\mathcal{D}}^{\ell_{\text{nll}}}(\mathbf{w}) \leq \\
&\underbrace{\frac{\|\boldsymbol{y}-f(\mathbf{X};\mathbf{w}_{\hat{\rho}})\|_2^2}{2n} + \left(\frac{1}{\frac{\lambda h}{d}+\frac{1}{\sigma_\pi^2}}\right)\frac{h}{2n} + \frac{1}{2}\ln(2\pi)}_{\text{Empirical Risk}} + \underbrace{\frac{\sigma_{\boldsymbol{x}}^2(\sigma_\pi^2 d + \|\mathbf{w}_* - \mathbf{w}_\pi\|_2^2)}{2-2\lambda n\sigma_{\boldsymbol{x}}^2\sigma_\pi^2} + \frac{\sigma_\epsilon^2}{2}}_{\text{Moment}} \\
&+ \underbrace{\frac{1}{\lambda n}\left[\frac{1}{2}\left(\frac{d}{\sigma_\pi^2}\frac{1}{\frac{\lambda h}{d}+\frac{1}{\sigma_\pi^2}} + \frac{1}{\sigma_\pi^2}\|\mathbf{w}_{\hat{\rho}} - \mathbf{w}_\pi\|_2^2 - d - d\ln\frac{1}{\frac{\lambda h}{d}+\frac{1}{\sigma_\pi^2}} + d\ln\sigma_\pi^2\right) + \ln\frac{1}{\delta}\right]}_{\text{KL}}.
\end{aligned} \tag{2}$$

To estimate the bound we need to measure the following quantities

- $h = \left[\sum_i \sum_j (\nabla_{\mathbf{w}} f(\boldsymbol{x}_i;\mathbf{w}_{\hat{\rho}})_j)^2\right]$ the curvature at the minimum. Note how this corresponds to the the sum of the squared gradients per data sample. We can compute this term using the Laplace-Redux package (Daxberger et al., 2021a) which has as a backend the BackPACK package (Dangel et al., 2019) or the ASDL package (Osawa, 2021).
- $\mathbf{w}_\pi$ and $\mathbf{w}_{\hat{\rho}}$ the prior and posterior means. We can also compute these terms explicitly. We typically train a deterministic neural network with SGD on $\mathcal{Z}_{\text{train}}$ to obtain a MAP estimate $\mathbf{w}_\pi$ then we also set $\mathbf{w}_{\hat{\rho}} = \mathbf{w}_\pi$, such that $\|\mathbf{w}_{\hat{\rho}} - \mathbf{w}_\pi\|_2^2 = 0$ in the KL term. This typically makes bounds tighter and is valid so long as we evaluate the other terms in the bound on $\mathcal{Z}_{\text{trainsuffix}}$.
- $\sigma_{\boldsymbol{x}}^2$ the per weight variance of the per-sample gradients. We estimate these using the data split reserved for approximating the full distribution $\mathcal{Z}_{\text{true}}$. We estimate this quantity as $\sigma_{\boldsymbol{x}}^2 = \frac{\sum_{i\in\mathcal{Z}_{\text{true}}}\sum_j(\nabla_{\mathbf{w}} f(\boldsymbol{x}_i;\mathbf{w}_{\hat{\rho}})_j)^2}{\#\mathcal{Z}_{\text{true}}\#\text{weights}}$. Note that we do the above instead of actually fitting a Gaussian mixture on the gradients which would be tedious and error prone.
- $\|\boldsymbol{y} - f(\mathbf{X};\mathbf{w}_{\hat{\rho}})\|_2^2$ the MSE of the MAP classifier.
- σ_ϵ^2 the aleatoric uncertainty of the data. While we could estimate this using for example a Gaussian Process, since it is just a small constant we set it to be $\sigma_\epsilon^2 = 1$ in all experiments
- $\|\mathbf{w}_* - \mathbf{w}_\pi\|_2^2$ the ℓ_2 norm of the difference between the weights of the oracle function that generated the labels $\mathbf{w}_*$, and our prior mean $\mathbf{w}_\pi$. The oracle quantity $\mathbf{w}_*$ is unknown. Setting $\|\mathbf{w}_* - \mathbf{w}_\pi\|_2^2 \approx \|\mathbf{w}_{\hat{\rho}} - \mathbf{w}_\pi\|_2^2 = \|\mathbf{w}_\pi - \mathbf{w}_\pi\|_2^2 = 0$ might be too optimistic so instead we set $\|\mathbf{w}_* - \mathbf{w}_\pi\|_2^2 \approx \|\mathbf{w}_{\hat{\rho}}\|_2^2 = \|\mathbf{w}_\pi\|_2^2$. (Remember that we set $\mathbf{w}_{\hat{\rho}} = \mathbf{w}_\pi$ to make the bound tighter.)

The values of the following variables can be set when evaluating the bound:

- σ_π^2 the prior variance.

- d the number of weights in the model.
- λ the temperature parameter.
- δ the confidence of the bound, we typically use $\delta = 0.05$.

B.4.2 MIXED BOUND EVALUATION $\mathcal{B}_{\text{mixed}}$

We need to evaluate the following bound

$$\begin{aligned}
&\mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\mathcal{L}_{\mathcal{D}}^{\ell_{\text{nll}}}(\mathbf{w}) \leq \\
&\underbrace{\frac{\|\boldsymbol{y} - f(\mathbf{X}; \mathbf{w}_{\hat{\rho}})\|_2^2}{2n} + \left(\frac{1}{\frac{\lambda h}{d} + \frac{1}{\sigma_\pi^2}}\right)\frac{h}{2n} + \frac{1}{2}\ln(2\pi)}_{\text{Empirical Risk}} \\
&+ \underbrace{\frac{1}{\lambda n}\ln \mathbf{E}_{f\sim\pi}\mathbf{E}_{(X',Y')\sim\mathcal{D}^n}\exp\left[\lambda n\left(\mathcal{L}_{\mathcal{D}}^{\ell_{\text{nll}}}(f) - \hat{\mathcal{L}}_{X',Y'}^{\ell_{\text{nll}}}(f)\right)\right]}_{\text{Moment}} \qquad (3) \\
&+ \underbrace{\frac{1}{\lambda n}\left[\frac{1}{2}\left(\frac{d}{\sigma_\pi^2}\frac{1}{\frac{\lambda h}{d} + \frac{1}{\sigma_\pi^2}} + \frac{1}{\sigma_\pi^2}\|\mathbf{w}_{\hat{\rho}} - \mathbf{w}_\pi\|_2^2 - d - d\ln\frac{1}{\frac{\lambda h}{d} + \frac{1}{\sigma_\pi^2}} + d\ln\sigma_\pi^2\right) + \ln\frac{1}{\delta}\right]}_{\text{KL}}
\end{aligned}$$

To estimate the bound we need to measure the same quantities as in the $\mathcal{B}_{\text{approximate}}$ except for $\sigma_{\boldsymbol{x}}^2, \sigma_\epsilon^2$ and $\|\mathbf{w}_* - \mathbf{w}_\pi\|_2^2$. In their place we need to estimate

$$\Psi_{\ell,\pi,\mathcal{D}}(\lambda, n) = \ln \mathbf{E}_{f\sim\pi}\mathbf{E}_{(X',Y')\sim\mathcal{D}^n}\exp\left[\lambda\left(\mathcal{L}_{\mathcal{D}}^{\ell_{\text{nll}}}(f) - \hat{\mathcal{L}}_{X',Y'}^{\ell_{\text{nll}}}(f)\right)\right].$$

We can approximate this term as

$$\Psi_{\ell,\pi,\mathcal{D}}(\lambda, n) \approx \ln\frac{1}{m}\sum_{f_i\sim\pi}\sum_{(X_j',Y_j')\sim\mathcal{D}^n}\exp\left[\lambda\left(\mathcal{L}_{\mathcal{D}}^{\ell_{\text{nll}}}(f_i) - \hat{\mathcal{L}}_{X_j',Y_j'}^{\ell_{\text{nll}}}(f_i)\right)\right]$$

by using Monte Carlo sampling. We note that this Moment term requires a sub-Gaussian or more generally a sub-Weibull assumption on the random variable $V = \left(\mathcal{L}_{\mathcal{D}}^{\ell_{\text{nll}}}(f_i) - \hat{\mathcal{L}}_{X_j',Y_j'}^{\ell_{\text{nll}}}(f_i)\right)$, so that we are sure that it is bounded. We use $m = 100$ samples to approximate this term in all experiments. Even when assuming that the variable V is sub-Gaussian or sub-Weibull (and therefore has light tails) the exponentiated variable might have heavy tails. More importantly for large values of λ the variance of the corresponding naive Monte Carlo estimator that we implement might have large or even infinite variance, making our empirical estimate unreliable. We thus present the results on the bounds for the regression data with these caveats in mind.

B.4.3 ALQUIER BOUND EVALUATION $\mathcal{B}_{\text{Alquier}}$

We need to evaluate the following bound

$$\begin{aligned}
&\mathbf{E}_{\mathbf{w}\sim\hat{\rho}}\mathcal{L}_{\mathcal{D}}^{\ell_{\text{nll}}}(\mathbf{w}) \leq \underbrace{\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}_{X,Y}^{\ell_{\text{nll}}}(f)}_{\text{Empirical Risk}} + \underbrace{\frac{1}{\lambda n}\ln \mathbf{E}_{f\sim\pi}\mathbf{E}_{(X',Y')\sim\mathcal{D}^n}\exp\left[\lambda n\left(\mathcal{L}_{\mathcal{D}}^{\ell_{\text{nll}}}(f) - \hat{\mathcal{L}}_{X',Y'}^{\ell_{\text{nll}}}(f)\right)\right]}_{\text{Moment}} \\
&+ \underbrace{\frac{1}{\lambda n}\left[\frac{1}{2}\left(\frac{d}{\sigma_\pi^2}\frac{1}{\frac{\lambda h}{d} + \frac{1}{\sigma_\pi^2}} + \frac{1}{\sigma_\pi^2}\|\mathbf{w}_{\hat{\rho}} - \mathbf{w}_\pi\|_2^2 - d - d\ln\frac{1}{\frac{\lambda h}{d} + \frac{1}{\sigma_\pi^2}} + d\ln\sigma_\pi^2\right) + \ln\frac{1}{\delta}\right]}_{\text{KL}}
\end{aligned} \qquad (4)$$

To estimate the bound we need to measure the same quantities as in the $\mathcal{B}_{\text{mixed}}$ except for the empirical risk. We estimate this by sampling directly from the empirical loss using Monte Carlo sampling

$$\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell}_{X,Y}(f) \approx \frac{1}{m}\sum_{f_i\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell}_{X,Y}(f_i).$$

We use $m = 100$ samples to approximate this term in all experiments.

B.4.4 ADDITIONAL NOTES ON BOUND EVALUATION

For the regression datasets We tested 20 different values for σ_π^2 in $[0.00001, 0.1]$.

We try to make our bounds as tight as possible. To do this we try to control the term $\|\mathbf{w}_{\hat{\rho}} - \mathbf{w}_\pi\|_2^2$ which typically dominates the bound. We follow for all tasks a variation of the approach of Dziugaite et al. (2021). Specifically we use $\mathcal{Z}_{\text{train}}$ to learn a prior mean $\mathbf{w}_\pi$ then we set, $\mathbf{w}_{\hat{\rho}} = \mathbf{w}_\pi$, such that $\|\mathbf{w}_{\hat{\rho}} - \mathbf{w}_\pi\|_2^2 = 0$. Note that we can still evaluate a valid bound so long as we set (X, Y) in Theorem 1 to be independent of the prior mean. This is the reason why we separated a part of the training set in the form of $\mathcal{Z}_{\text{validation}}$. We thus set $(X, Y) = \mathcal{Z}_{\text{validation}}$ in Theorem 1. All bounds ($\mathcal{B}_{\text{Alquier}}$, $\mathcal{B}_{\text{mixed}}$, $\mathcal{B}_{\text{approximate}}$) can then be evaluated by taking into account this substitution. Note that our final model deviates from what would be typically used in practice, but it shouldn't deviate significantly. Specifically our models are a modification of the commonly used Laplace approximation (Daxberger et al., 2021a). We only use $(X, Y) = \mathcal{Z}_{\text{validation}}$ to learn the *posterior variance* of a Laplace approximation, and in particular to estimate the curvature parameter h.

For most datasets (such as CIFAR-10) we are not aware of extended versions, and thus we would necessarily have to draw $\mathcal{Z}_{\text{true}}$ from $\mathcal{Z}_{\text{train}}$. This is why we cannot estimate the Alquier bound for CIFAR-10, CIFAR-100, SVHN and FMNIST, because to estimate the Moment term we would need a large $\mathcal{Z}_{\text{true}}$ which would necessarily limit the size of the training set significantly. Note that since in Moment term we have to draw $X', Y' \sim \mathcal{D}$ of size $|(X', Y')| = n = |\mathcal{Z}_{\text{validation}}|$ we would need $|\mathcal{Z}_{\text{true}}| \gg |\mathcal{Z}_{\text{validation}}|$ so the decrease in the available training samples would be significant, and consequently also the relevance for real applications.

In our experiments we test multiple values of λ and σ_π^2. Typically one would need to take a union bound over a grid on these parameters so as for the generalization bound to be valid (Dziugaite & Roy, 2017). However this typically costs only logarithmically to the actual bound. We ignore these calculations as our bounds are in general quite loose anyway, and these calculations would result in additional terms would make the final bound even more complex.

For the bounds to be valid, one would typically want to show concentration inequalities such that the Monte Carlo estimates of the Empirical Risk and the Moment terms concentrate close to the true expected value with high probability. We do not provide such guarantees. Note however that, at least for the Empirical Risk term, our sample size of $m = 100$ from the posterior distribution over weights is a sample size that is typically used in practice and provides good estimates. Regarding the Moment term we typically use $m = 100$. Specifically we sample 10 samples $\mathbf{w}_i \sim \pi$ and for each $\mathbf{w}_i$ we sample 10 samples from $X_j, Y_j \sim \mathcal{D}$. We have tried to balance sampling sufficiently to approximate the expectation on the one hand, and also not too much such that the computations become prohibitive.

B.5 ADDITIONAL REGRESSION RESULTS

We find ten MAP estimates for the neural network weights of the Abalone, Diamonds and KC_House datasets by training on Z_{train} using Stochastic Gradient Descent (SGD) with stepsize $\eta = 10^{-3}$ for ten epochs. We then fit an Isotropic Laplace approximation to each MAP estimate using $Z_{\text{validation}}$. For different values of λ we then estimate the Alquier bound (equation 4) using $X, Y = Z_{\text{validation}}$, as well as the *test* NLL of the posterior predictive on Z_{test}. We take a grid over prior variances σ_π^2, and we present results for $\sigma_\pi^2 = 0.005$ although the behaviour is similar for the other prior values.

We plot the results for all datasets in Figure 1. Somewhat surprisingly, the test NLL always decreases with colder posteriors up to the point where the classifier is essentially deterministic. The $\mathcal{B}_{\text{Alquier}}$ bound correlates tightly with this behaviour. These results are somewhat surprising, in that we would expect there to be a minimum in the curves, such that *some* posterior variance $\sigma_{\hat{\rho}} \geq 0$ gives better test results than the MAP estimate. These results could be due to the poor (Isotropic) approximation

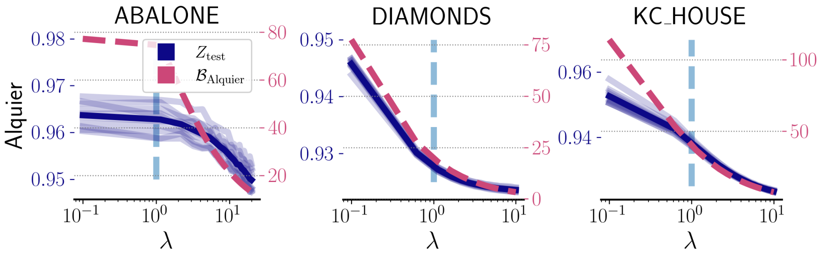


Figure 1: $\mathcal{B}_{\text{Alquier}}$ PAC-Bayes bound – – – and test NLL ━━ mean, as well as 10 MAP trials ━━ (we denote $\lambda = 1$ by – – –). For varying λ for the regression tasks on the UCI Abalone, UCI Diamonds and KC_House datasets. $\mathcal{B}_{\text{Alquier}}$ bound closely tracks the test NLL. There is a rapid improvement as $\lambda \uparrow$ followed by a slowdown in improvements. Coldest posteriors $\lambda \gg 1$ are always best.

to the posterior. We furthermore note the caveats mentioned in the B.4.2 which make the estimates of the Moment term unreliable.

In Figure 7 we see more detailed experiments on the regression datasets Abalone, Diamonds and KC_House. We see that for all the neural networks that we trained across all datasets the $\mathcal{B}_{\text{approximate}}$ bound is very loose. Specifically for the cases we consider λ is always restricted to be $\lambda < 1$ which is very limiting since we want to investigate cold posteriors and $\lambda > 1$. When comparing the $\mathcal{B}_{\text{mixed}}$ and $\mathcal{B}_{\text{Alquier}}$ bounds we see that there is little change in the bound values. Specifically estimating the Empirical Risk with Monte Carlo sampling, instead of using a Taylor expansion of second order (as in $\mathcal{B}_{\text{approximate}}$) doesn't yield significant benefits. The big improvements are the result of estimating the Moment term using Monte Carlo sampling.

B.6 COMPLETE CLASSIFICATION RESULTS

We plot in Figure 4 the standard Isotropic and standard KFAC cases for the ECE. Even without data augmentation and even when we optimize the prior variance using the marginal likelihood, we find that all three cases of temperatures (cold posterior, warm posterior, as well as posterior with $\lambda = 1$) can be optimal, for varying datasets. Unfortunately we are not aware of approaches to directly bound the ECE. The ECE is notable for having a significantly different behaviour from the NLL and the 0-1 Loss. At the same time, better calibration in terms of ECE than a simple MAP estimate is one of the purported main benefits of the Bayesian paradigm.

In Figure 5 we plot the Pareto front of the *test* 0-1 Loss with respect to the *test* ECE. The top row is the standard Isotropic case and the bottom row is the standard KFAC case. We see that in most cases there is a clear tradeoff between the test 0-1 Loss and the test ECE. These results might be relevant for the applicability of the Laplace approximation for improving the ECE.

In Figure 6 we see that data augmentation (random flips and crops) results in better test accuracy and makes the matrix inversion in the Laplace approximation better posed such that the accuracy on CIFAR-100 is within a normal range.

B.7 ADDITIONAL RESULTS ON BOUND TERMS

We now present additional results on the behaviour of the different terms (Empirical Risk, KL, Moment) of the different bounds ($\mathcal{B}_{\text{approximate}}$, $\mathcal{B}_{\text{mixed}}$, $\mathcal{B}_{\text{Alquier}}$). We plot the results in Figure 8. We see that across all cases the $\mathcal{B}_{\text{approximate}}$ bound is significantly off scale in the x-axis. For our realistic choices of σ_π^2 the parameter λ is restricted to be $\lambda < 1$, which is very limiting for the setting we want to investigate $\lambda \geq 1$. However the bound gives some useful intuition regarding how the terms vary as we change λ. Across all datasets the Empirical Risk decreases as we increase λ, the KL term also increases, while the Moment term increases, but with a much slower rate than is implied by $\mathcal{B}_{\text{approximate}}$. Comparing the $\mathcal{B}_{\text{mixed}}$ and $\mathcal{B}_{\text{Alquier}}$ bounds we see that we get a significantly tighter estimate of the Empirical Risk. However this doesn't improve significantly the bound, in that the KL term is orders of magnitude larger than the other terms. Contrary to the main text we plot the

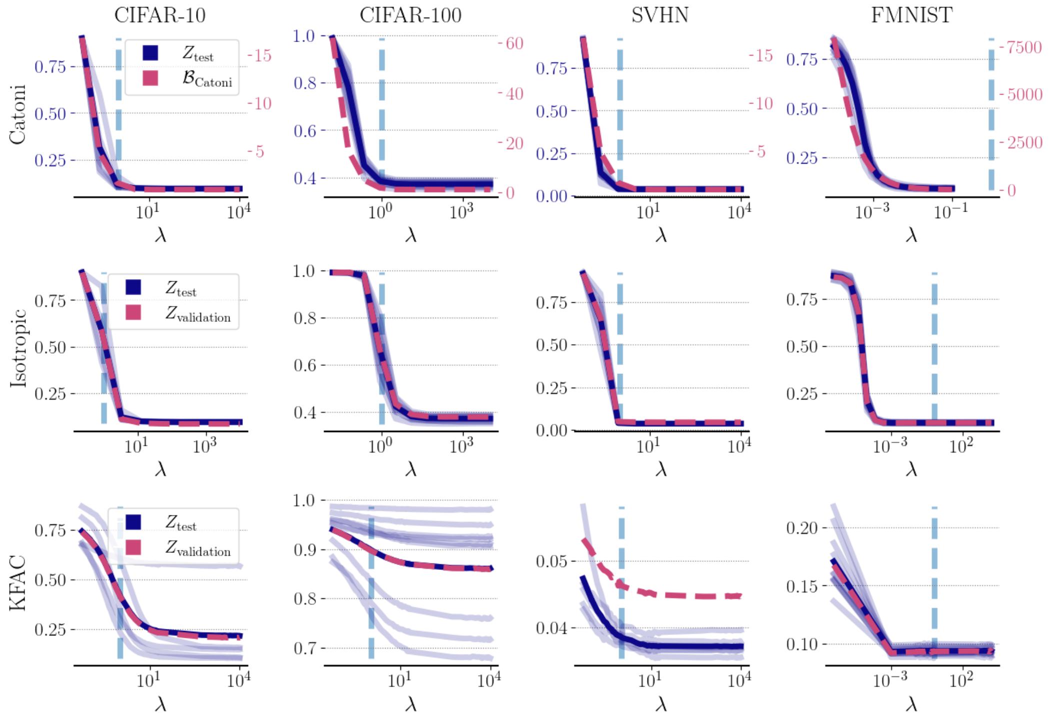


Figure 2: Test 0-1 Loss ━━ mean, as well as 10 MAP trials ━━, along with the generalization certificate ╍╍ (we denote $\lambda = 1$ by ╍╍): $\mathcal{B}_{\text{Catoni}}$ PAC-Bayes bound (top), standard Isotropic Laplace posterior (middle) and standard KFAC (bottom). The $\mathcal{B}_{\text{Catoni}}$ PAC-Bayes bound closely tracks the test 0-1 Loss. For the standard Isotropic and KFAC posteriors the test and validation 0-1 Loss behave similar to the Catoni case, with a rapid improvement as $\lambda \uparrow$ followed by a plateau. Coldest posteriors $\lambda \gg 1$ are always best.

original values for all quantities but on logarithmic scale when necessary. (In the main text we first normalize the KL and then add it to the other terms, which gives a result that is a little more difficult to interpret). The results for each dataset are for a random choice of σ_π^2.

C PROOF OF THEOREM 1

We include here a proof of Theorem 1, first presented in Germain et al. (2016), and based on Alquier et al. (2016); Bégin et al. (2016); Germain et al. (2016) to illustrate how the Moment term and the temperature parameter λ arise in the final bound. The Donsker–Varadhan's change of measure states that, for any measurable function $\phi : \mathcal{F} \to \mathbb{R}$, we have

$$\mathbf{E}_{f\sim\hat{\rho}}\phi(f) \leq \mathrm{KL}(\hat{\rho}\|\pi) + \ln(\mathbf{E}_{f\sim\pi}\exp[\phi(f)]).$$

Thus, with $\phi(f) := \lambda\left(\mathcal{L}^{\ell}_{\mathcal{D}}(f) - \hat{\mathcal{L}}^{\ell}_{X,Y}(f)\right)$, we obtain $\forall\hat{\rho}$ on $\mathcal{F}$:

$$\begin{aligned}\lambda\left(\mathbf{E}_{f\sim\hat{\rho}}\mathcal{L}^{\ell}_{\mathcal{D}}(f) - \mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell}_{X,Y}(f)\right) &= \mathbf{E}_{f\sim\hat{\rho}}\lambda\left(\mathcal{L}^{\ell}_{\mathcal{D}}(f) - \hat{\mathcal{L}}^{\ell}_{X,Y}(f)\right)\\ &\leq \mathrm{KL}(\hat{\rho}\|\pi) + \ln\left(\mathbf{E}_{f\sim\pi}\exp[\lambda\left(\mathcal{L}^{\ell}_{\mathcal{D}}(f) - \hat{\mathcal{L}}^{\ell}_{X,Y}(f)\right)]\right).\end{aligned}$$

Now, we apply Markov's inequality on the random variable $\zeta_\pi(X,Y) := \mathbf{E}_{f\sim\pi}\exp\left[\lambda\left(\mathcal{L}^{\ell}_{\mathcal{D}}(f) - \hat{\mathcal{L}}^{\ell}_{X,Y}(f)\right)\right]$ and get

$$\Pr_{(X,Y)\sim\mathcal{D}^n}\left(\zeta_\pi(X,Y) \leq \frac{1}{\delta}\mathbf{E}_{(X',Y')\sim\mathcal{D}^n}\zeta_\pi(X',Y')\right) \geq 1-\delta.$$

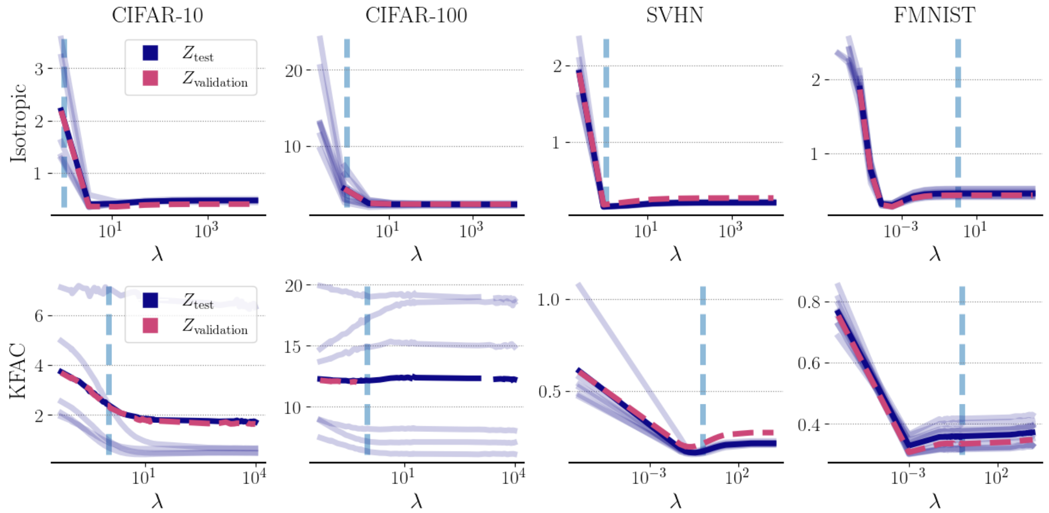


Figure 3: Test NLL ━━ mean, as well as 10 MAP trials ━━, along with the validation NLL ╍╍ (we denote $\lambda = 1$ by ╍╍) for the Standard Isotropic Laplace posterior (top) and standard KFAC (bottom). The test and validation NLL show warm posteriors (FMNIST and SVHN KFAC), cold posteriors (CIFAR-10) and posteriors with $\lambda = 1$ (SVHN Isotropic). The general trend remains a rapid improvement as $\lambda \uparrow$ followed by a plateau, however the coldest posteriors $\lambda \gg 1$ are not always optimal contrary to the 0-1 Loss case.

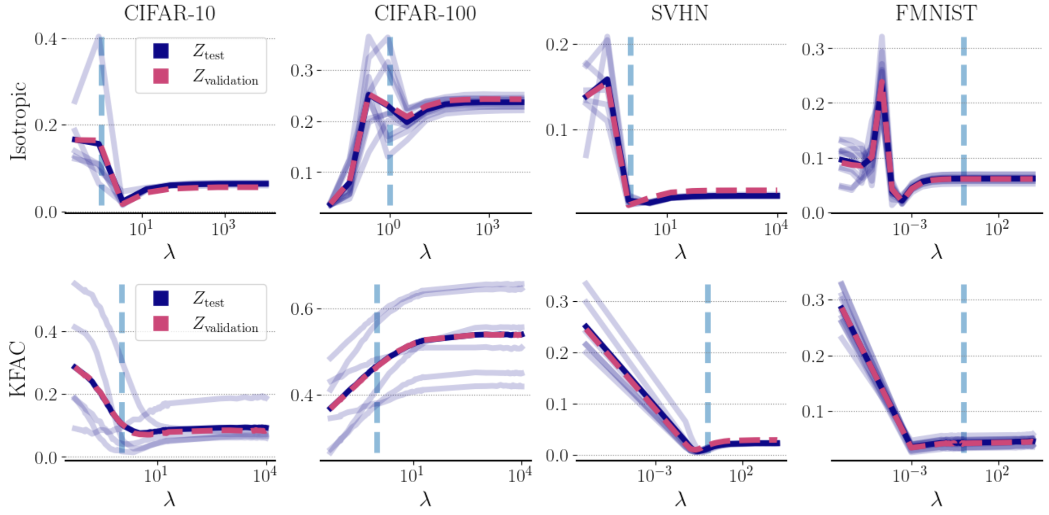


Figure 4: Test ECE ━━ mean, as well as 10 MAP trials ━━, along with the validation ECE ╍╍ (we denote $\lambda = 1$ by ╍╍) for the Standard Isotropic Laplace posterior (top) and standard KFAC (bottom). The test and validation ECE show warm posteriors (FMNIST and SVHN KFAC), cold posteriors (CIFAR-10) and posteriors with $\lambda = 1$ (SVHN Isotropic). The general trend remains a rapid improvement as $\lambda \uparrow$ followed by a plateau, however the coldest posteriors $\lambda \gg 1$ are not always optimal contrary to the 0-1 Loss case.

This implies that with probability at least $1 - \delta$ over the choice $(X, Y) \sim \mathcal{D}^n$, we have $\forall \hat{\rho}$ on $\mathcal{F}$

$$\mathbf{E}_{f\sim\hat{\rho}}\mathcal{L}^{\ell}_{\mathcal{D}}(f) \leq \mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell}_{X,Y}(f)+\frac{1}{\lambda}\left[\mathrm{KL}(\hat{\rho}\|\pi) + \ln\frac{\mathbf{E}_{(X',Y')\sim\mathcal{D}^n}\mathbf{E}_{f\sim\pi}\exp[\lambda\left(\mathcal{L}^{\ell}_{\mathcal{D}}(f)-\hat{\mathcal{L}}^{\ell}_{X,Y}(f)\right)]}{\delta}\right].$$

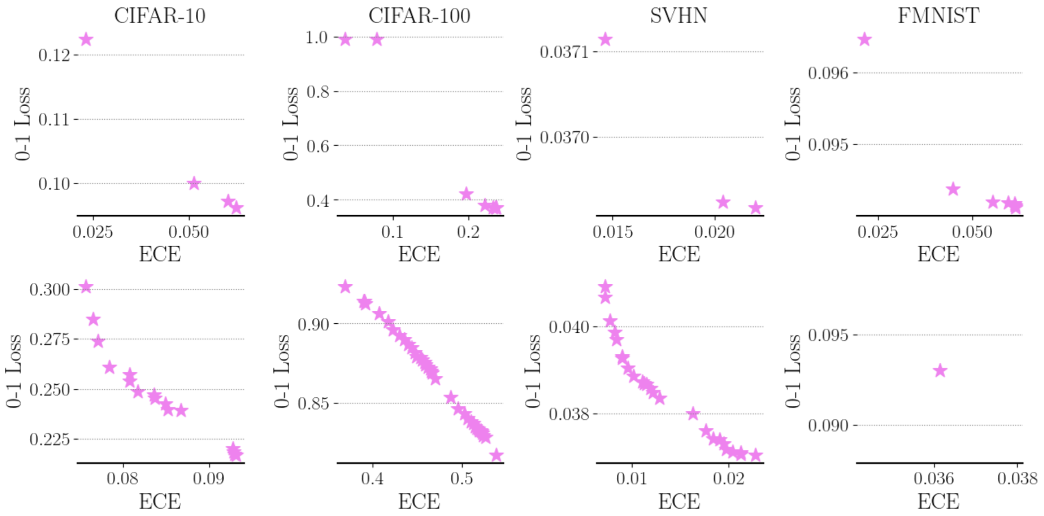


Figure 5: We plot the Pareto front of the *test* 0-1 Loss with respect to the *test* ECE. The top row is the standard Isotropic case and the bottom row is the standard KFAC case. We see that in most cases there seems to be a tradeoff between the test 0-1 Loss and the test ECE.

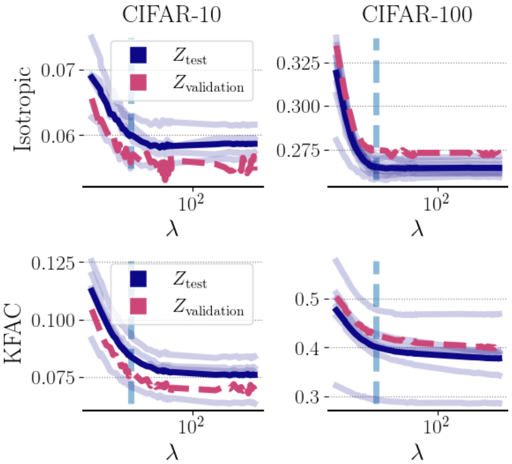
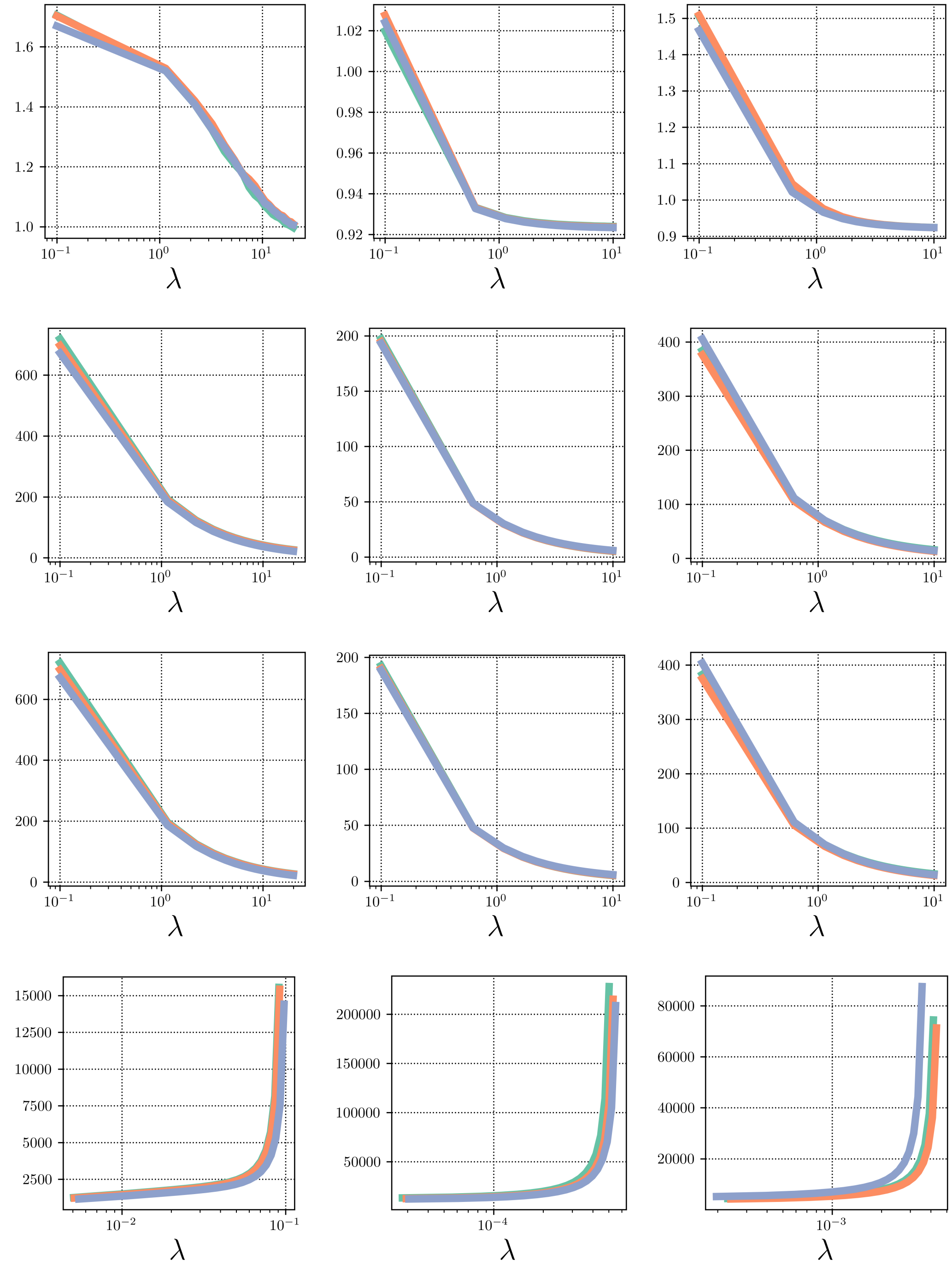


Figure 6: Test 0-1 Loss ━━━ mean, as well as 10 MAP trials ━━━, along with the validation 0-1 Loss – – – (we denote $\lambda = 1$ by – – –) for the Standard Isotropic Laplace posterior (top) and standard KFAC (bottom) for CIFAR-10 and CIFAR-100 with data augmentation (random flips and crops). The performance on both improves significantly and the Laplace approximation becomes better posed.

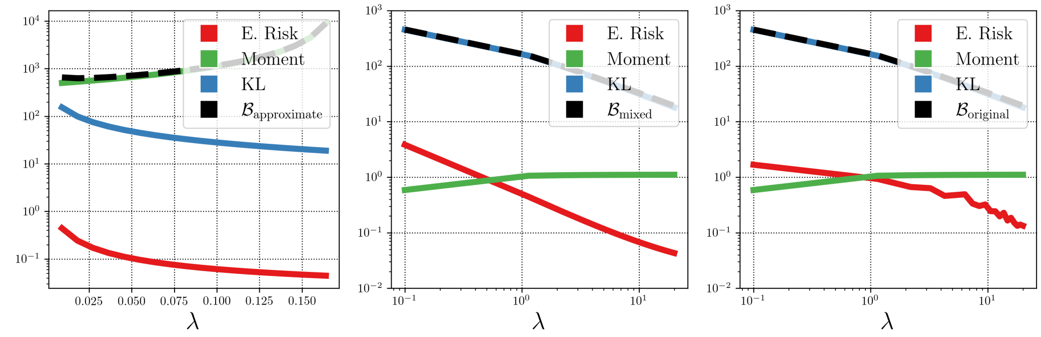


(a) UCI, Abalone

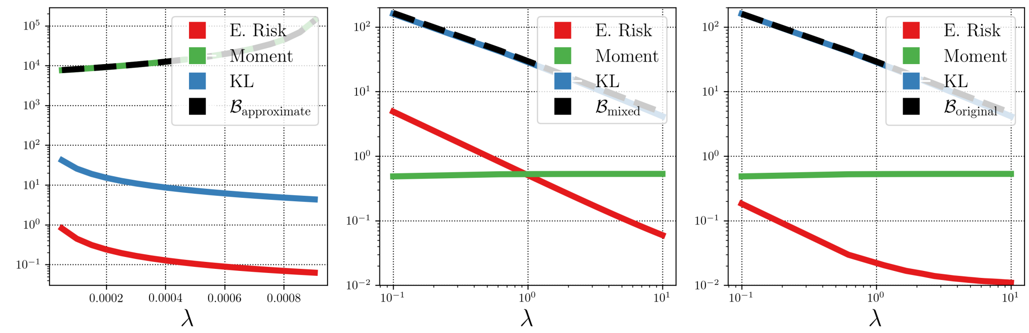
(b) UCI, Diamonds

(c) KC_House

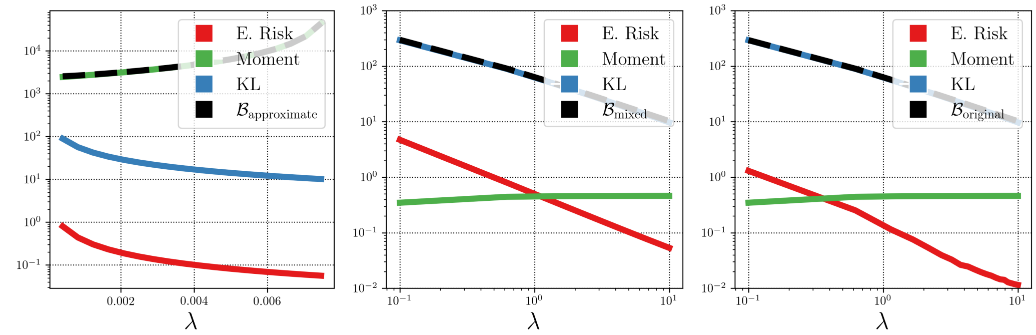
Figure 7: $\sigma_\pi^2 = 0.04737$. First row: Negative log-likelihood. Second row: PAC-Bayes Alquier bound $\mathcal{B}_{\text{Alquier}}$. Third row: PAC-Bayes mixed bound $\mathcal{B}_{\text{mixed}}$. Fourth row: PAC-Bayes approximate bound $\mathcal{B}_{\text{approximate}}$ for varying λ. Different colours correspond to different MAP estimates. All quantities show very little variation around their mean, hence the significant overlap.



(a) Abalone



(b) Diamonds



(c) KC_House

Figure 8: The different terms of the $\mathcal{B}_{\text{approximate}}$ (left column), $\mathcal{B}_{\text{mixed}}$ (middle column), and $\mathcal{B}_{\text{Alquier}}$ (right column) bounds. We see that the $\mathcal{B}_{\text{approximate}}$ bound gives some useful intuition. Moving from $\mathcal{B}_{\text{mixed}}$ to $\mathcal{B}_{\text{Alquier}}$ gives some improvements to the Empirical Risk values but not to the bound values which are orders of magnitude larger.

D FAQ

- *What is the purpose of $\mathcal{Z}_{\text{true}}$ set?* In the Alquier bound we need to compute the Moment $\Psi_{\ell,\pi,\mathcal{D}}(\lambda, n) = \ln \mathbf{E}_{f\sim\pi}\mathbf{E}_{X',Y'\sim\mathcal{D}^n} \exp\left[\lambda n \left(\mathcal{L}^{\ell}_{\mathcal{D}}(f) - \hat{\mathcal{L}}^{\ell}_{X',Y'}(f)\right)\right]$. To estimate the Moment we do Monte Carlo sampling $f \sim \pi$ and $X', Y' \sim \mathcal{D}^n$. We use the $\mathcal{Z}_{\text{true}}$ set to sample $X', Y' \sim \mathcal{D}^n$.
- *How is the case where you learn the prior and posterior mean using the $\mathcal{Z}_{\text{train}}$ and then the posterior variance using $\mathcal{Z}_{\text{validation}}$ related to the standard Laplace approximation/Variational Inference?* Our case can be seen as a greatly simplified case of Online Variational Inference Chérief-Abdellatif et al. (2019) for the set $\mathcal{Z}_{\text{validation}} \cup \mathcal{Z}_{\text{train}}$. In fact in a truly Bayesian approach we would typically optimize with $\mathcal{Z}_{\text{validation}} \cup \mathcal{Z}_{\text{train}}$ as the posterior is assumed to reflect our best guess after seeing the data, making a validation set redundant. We include the Standard Isotropic and standard KFAC case (where $\mathcal{Z}_{\text{validation}}$ is not used for training but simply to provide a generalization certificate) so as to demonstrate that the behaviour of our approach is relevant for standard practice.
- *Isn't the fact that $\lambda \gg 1$ well known in the PAC-Bayes literature?* We are aware of results such as the one in Catoni (2007) p13 where for *fixed* prior and posterior distributions the optimal λ is shown to be approximately $\lambda = \sqrt{\frac{2a(\mathrm{KL}(\hat{\rho}||\pi) - \log(\epsilon))}{n\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell}_{X,Y}(f)(1-\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell}_{X,Y}(f))}}$ for $a > 1$ (note the change in the scaling of λ to match our own text). Taking this in to account, for small KL the value of λ will be through this analysis most likely less than 1. More importantly, the relevant setting for the cold-posterior effect is the one where we *optimize the posterior* for different values of λ, and not for fixed posteriors which is the setting of Catoni (2007). In particular it is not obvious that the result of Catoni (2007) is the same when changing $\hat{\rho}$ based on λ.
- *Can you explain the Gradients as Gaussian mixture: $\nabla_{\mathbf{w}} f(\boldsymbol{x}; \mathbf{w}_{\hat{\rho}}) \sim \sum_{i=1}^{k} \phi_i \mathcal{N}(\boldsymbol{\mu}_i, \sigma^2_{\boldsymbol{x}i}\mathbf{I})$ assumption?* The gradients per sample $\nabla_{\mathbf{w}} f(\boldsymbol{x}; \mathbf{w}_{\hat{\rho}})$ act as a non-linear feature vector for each $\boldsymbol{x}$. When the linearization of a neural network is plausible (and therefore the neural network is a linear classifier for high-dimensional feature vectors) it is also plausible that the generative model of the feature vectors of the data samples is a Gaussian mixture (see for example Bishop (2006) Section 4.2 for a discussion of Probabilistic Generative Models). Note that for *trained* neural networks, previous works have also shown that per sample gradients with respect to the weights, at $\mathbf{w}_{\hat{\rho}}$, are clusterable (Zancato et al., 2020) further supporting that the gradients of all the samples can be seen as a Gaussian mixture. When analyzing minima of the loss landscape (as we do here) linearization is reasonable even without assuming infinite width Zancato et al. (2020); Maddox et al. (2021).
- *Wouldn't the results be different if you optimized the ELBO to find MAP estimates? The ELBO would force the MAP minima to be flat and the "noise" from the posterior would affect less the test accuracy.* We use weight decay in our SGD implementation which should regularize somewhat our learned network. Furthermore when explicitly penalizing for the minima curvature Foret et al. (2020) researchers observe a consistent but overall small improvement compared to standard SGD. This leads us to believe that optimizing the ELBO and then computing the Laplace approximation would not significantly alter our results.
- *Hasn't the Laplace approximation been benchmarked before? What is the relationship with your experiments?* We are aware of at least the following works that benchmark the Laplace approximation (Daxberger et al., 2021a; Ritter et al., 2018; Antorán et al., 2022; Daxberger et al., 2021b; Immer et al., 2021). In Daxberger et al. (2021a) p23 Figure 8 (part of the Appendix) it is evident that when trying to fit the Laplace approximation over all the weights in the neural network there is some deterioration of the test accuracy with a corresponding improvement in AUROC. Even if for some MAP estimates fitting the Laplace improves both the accuracy and the AUROC, on average the Laplace accuracy is as good as the average MAP accuracy. In Ritter et al. (2018) p15 Tables 1 and 2 (part of the Appendix) we see that the accuracy is in both MNIST and CIFAR-100 cases slightly worse than the MAP accuracy. In Immer et al. (2021) p28 Table B4 (part of the Appendix) the difference between the best Laplace and the MAP estimate in terms of test accuracy is

on the order of 0.1% or even 0.01% and the gains in terms of ECE and OD-AUC are not consistent. In Antorán et al. (2022) p26 Figure 13 (part of the Appendix) the smallest prior variance is the best in terms of test NLL. Finally in Daxberger et al. (2021b) p15 Tables 2 and 15 (part of the Appendix) for the cases without corruptions, both in the MNIST and CIFAR-10 case the proposed Laplace approximation (over a subsample of the weights) results in lower test accuracy, though in the case of CIFAR-10 with gains in the ECE.

REFERENCES

Pierre Alquier, James Ridgway, and Nicolas Chopin. On the properties of variational approximations of Gibbs posteriors. *The Journal of Machine Learning Research*, 17(1):8374–8414, 2016.

Javier Antorán, David Janz, James U Allingham, Erik Daxberger, Riccardo Rb Barbano, Eric Nalisnick, and José Miguel Hernández-Lobato. Adapting the linearised laplace model evidence for modern deep learning. In *International Conference on Machine Learning*, pp. 796–821. PMLR, 2022.

Luc Bégin, Pascal Germain, François Laviolette, and Jean-Francis Roy. Pac-bayesian bounds based on the rényi divergence. In *Artificial Intelligence and Statistics*, pp. 435–444. PMLR, 2016.

Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

Olivier Catoni. *PAC-Bayesian supervised classification: the thermodynamics of statistical learning*, volume 56 of *Monograph Series*. Institute of Mathematical Statistics Lecture Notes, 2007.

Badr-Eddine Chérief-Abdellatif, Pierre Alquier, and Mohammad Emtiyaz Khan. A generalization bound for online variational inference. In *Asian conference on machine learning*, pp. 662–677. PMLR, 2019.

Felix Dangel, Frederik Kunstner, and Philipp Hennig. Backpack: Packing more into backprop. In *International Conference on Learning Representations*, 2019.

Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace Redux-Effortless Bayesian Deep Learning. *Advances in Neural Information Processing Systems*, 34, 2021a.

Erik Daxberger, Eric Nalisnick, James U Allingham, Javier Antorán, and José Miguel Hernández-Lobato. Bayesian deep learning via subnetwork inference. In *International Conference on Machine Learning*, pp. 2510–2521. PMLR, 2021b.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *Uncertainty in Artificial Intelligence*, 2017.

Gintare Karolina Dziugaite, Kyle Hsu, Waseem Gharbieh, Gabriel Arpino, and Daniel Roy. On the role of data in PAC-Bayes. In *International Conference on Artificial Intelligence and Statistics*, pp. 604–612. PMLR, 2021.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2020.

Pascal Germain, Francis Bach, Alexandre Lacoste, and Simon Lacoste-Julien. PAC-Bayesian theory meets Bayesian inference. *Advances in Neural Information Processing Systems*, 29, 2016.

harlfoxem. Kaggle. `https://www.kaggle.com/datasets/harlfoxem/housesalesprediction`, 2014.

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

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Citeseer*, 2009.

Frederik Kunstner, Philipp Hennig, and Lukas Balles. Limitations of the empirical fisher approximation for natural gradient descent. *Advances in neural information processing systems*, 32, 2019.

Fabian Küppers, Jan Kronenberger, Jonas Schneider, and Anselm Haselhoff. Bayesian confidence calibration for epistemic uncertainty modelling. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, July 2021.

Wesley Maddox, Shuai Tang, Pablo Moreno, Andrew Gordon Wilson, and Andreas Damianou. Fast adaptation with linearized neural networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 2737–2745. PMLR, 2021.

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

Kazuki Osawa. Asdl: Automatic second-order differentiation (for fisher, gradient covariance, hessian, jacobian, and kernel) library, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable Laplace approximation for neural networks. In *6th International Conference on Learning Representations*, volume 6. International Conference on Representation Learning, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arxiv*, 2017.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.

Luca Zancato, Alessandro Achille, Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Predicting training time without training. *Advances in Neural Information Processing Systems*, 33: 6136–6146, 2020.

Hongyi Zhang, Yann N Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. In *International Conference on Learning Representations*, 2019.