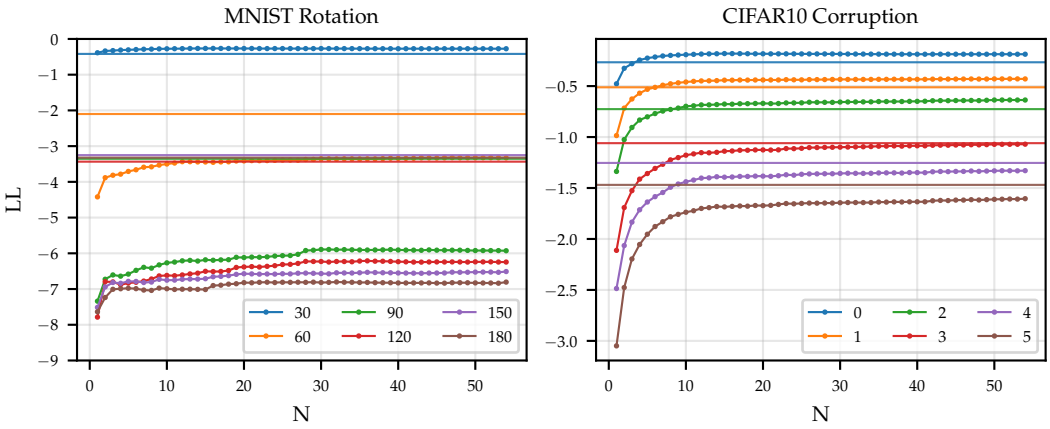

Figure 6. Rotated MNIST (left) and Corrupted CIFAR10 (right) results for deep ensembles (Lakshminarayanan et al., 2017) with large numbers of ensemble members (i.e. up to 55). Horizontal axis denotes number of ensemble members, and vertical axis denotes performance in terms of log-likelihood. Straight horizontal lines correspond to the performance of our method, as a reference. Colors denote different levels of rotation (left) and corruption (right).

A. Additional Image Classification Results

In this appendix, we provide additional experimental results for image classification tasks.

A.1. Comparing the Parameter Efficiency of Subnetwork Linearized Laplace with Deep Ensembles

Despite, the promising results shown by Subnetwork Linearized Laplace in Section 6.3, we note that our method has a notably larger space complexity than our baselines. We therefore investigate the parameter efficiency of our method.

Our ResNet18 Model has $\sim$11.2M parameters. Our subnetwork's covariance matrix contains $42{,}438^2$ parameters. This totals $\sim$1,830M parameters. This same amount of memory could be used to store around 163 ensemble elements. In Fig. 6 we compare our subnetwork Linearized Laplace model with increasingly large ensembles on both rotated MNIST and corrupted CIFAR10. Although the performance of ensembles improves as more networks are added, it plateaus around 15 ensemble elements. This is in agreement with the findings of recent works (Antorán et al., 2020; Ashukha et al., 2020; Lobacheva et al., 2020). At large rotations and corruptions, the log likelihood obtained by Subnetwork Linearised Laplace is greater than the asymptotic value obtained by ensembles. This suggests that using a larger number of parameters in a approximate posterior covariance matrix is a more efficient use of space than saving a large number of ensemble elements. We also note that inference in a very large ensemble requires performing a forward pass for every ensemble element. On the other hand, Linearised Laplace requires performing one backward pass for every output dimension and one forward pass.

A.2. Scalability of Subnetwork Linearised Laplace in the number of Weights

The aim of subnetwork inference is to scale existing posterior approximations to large networks. To further validate that this objective can be achieved, we perform subnetwork inference in ResNet50. We use a similar (slightly smaller) subnetwork size than we used with ResNet18: our subnetwork contains 39,190 / 23,466,560 (0.167%) parameters. The results obtained with this model are displayed in Fig. 7. Subnetwork inference in ResNet50 improves upon a simple MAP estimate of the weights in terms of both log-likelihood and calibration metrics.

A.3. Out-of-Distribution Rejection

In this section we provide additional results on out-of-distribution (OOD) rejection using predictive uncertainty. First, we train our models on a source dataset. We then evaluate them on the test set from our source dataset and on the test set of a target (out-of-distribution) dataset. We expect predictions for the target dataset to be more uncertain than those for the source dataset. Using predictive uncertainty as the discriminative variable we compute the area under ROC for each method under consideration and display them in Table 1. The CIFAR-SVHN and MNIST-Fashion dataset pairs are chosen following Nalisnick et al. (2019). On the CIFAR-SVHN task, all methods perform similarly, except for ensembles, which clearly does best. On MNIST-Fashion, SWAG performs best, followed by Subnetwork Linearised Laplace and ensembles.

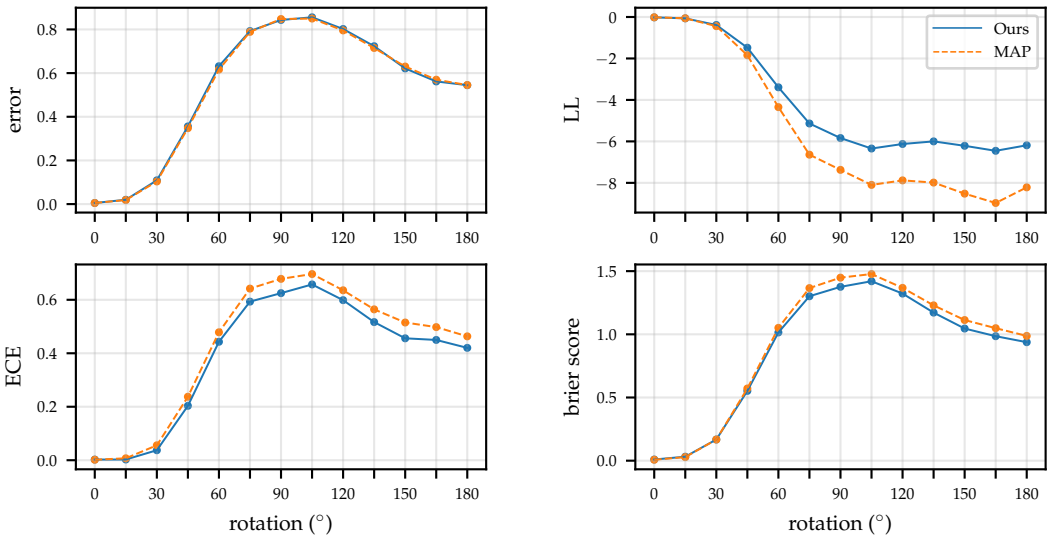

Figure 7. MNIST rotation results for ResNet-50, reporting predictive error, log-likelihood (LL), expected calibration error (ECE) and brier score. We choose a subnetwork containing only 0.167% (39,190 / 23,466,560) of the parameters of the full network.

Table 1. AUC-ROC scores for out-of-distribution detection, using CIFAR10 vs SVHN and MNIST vs FashionMNIST as in- (source) and out-of-distribution (target) datasets, respectively.

SOURCE	TARGET	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG
CIFAR10	SVHN	0.85±0.03	0.86±0.02	0.85±0.01	0.86±0.02	0.91±0.00	0.86±0.02	0.83±0.00
MNIST	Fashion	0.92±0.05	0.75±0.02	0.82±0.12	0.75±0.01	0.90±0.09	0.72±0.03	0.97±0.01

We also simulate a realistic OOD rejection scenario (Filos et al., 2019) by jointly evaluating our models on an in-distribution and an OOD test set. We allow our methods to reject increasing proportions of the data based on predictive entropy before classifying the rest. All predictions on OOD samples are treated as incorrect. Following (Nalisnick et al., 2019), we use CIFAR10 vs SVHN and MNIST vs FashionMNIST as in- and out-of-distribution datasets, respectively. Note that the SVHN test set is randomly sub-sampled down to a size of 10,000 to match that of CIFAR10. The results are shown in Fig. 8. On CIFAR-SVHN all methods perform similarly, with exceptions being ensembles, which perform best and SWAG which does worse. On MNIST-Fashion SWAG performs best, followed by Subnetwork Linearised Laplace. All other methods fail to distinguish very uncertain in-distribution data from low uncertainty OOD points.

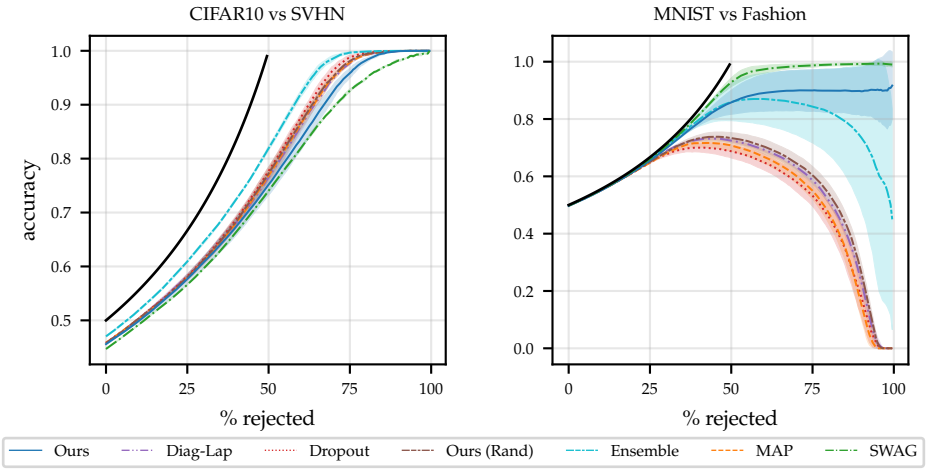

Figure 8. Rejection-classification plots.

A.4. Additional Rotation and Corruption Results

We complement our results from Fig. 4 in the main text with results on additional calibration metrics: ECE and Brier Score, in Fig. 9. Please refer to the appendix of (Antorán et al., 2020) for a description of these.

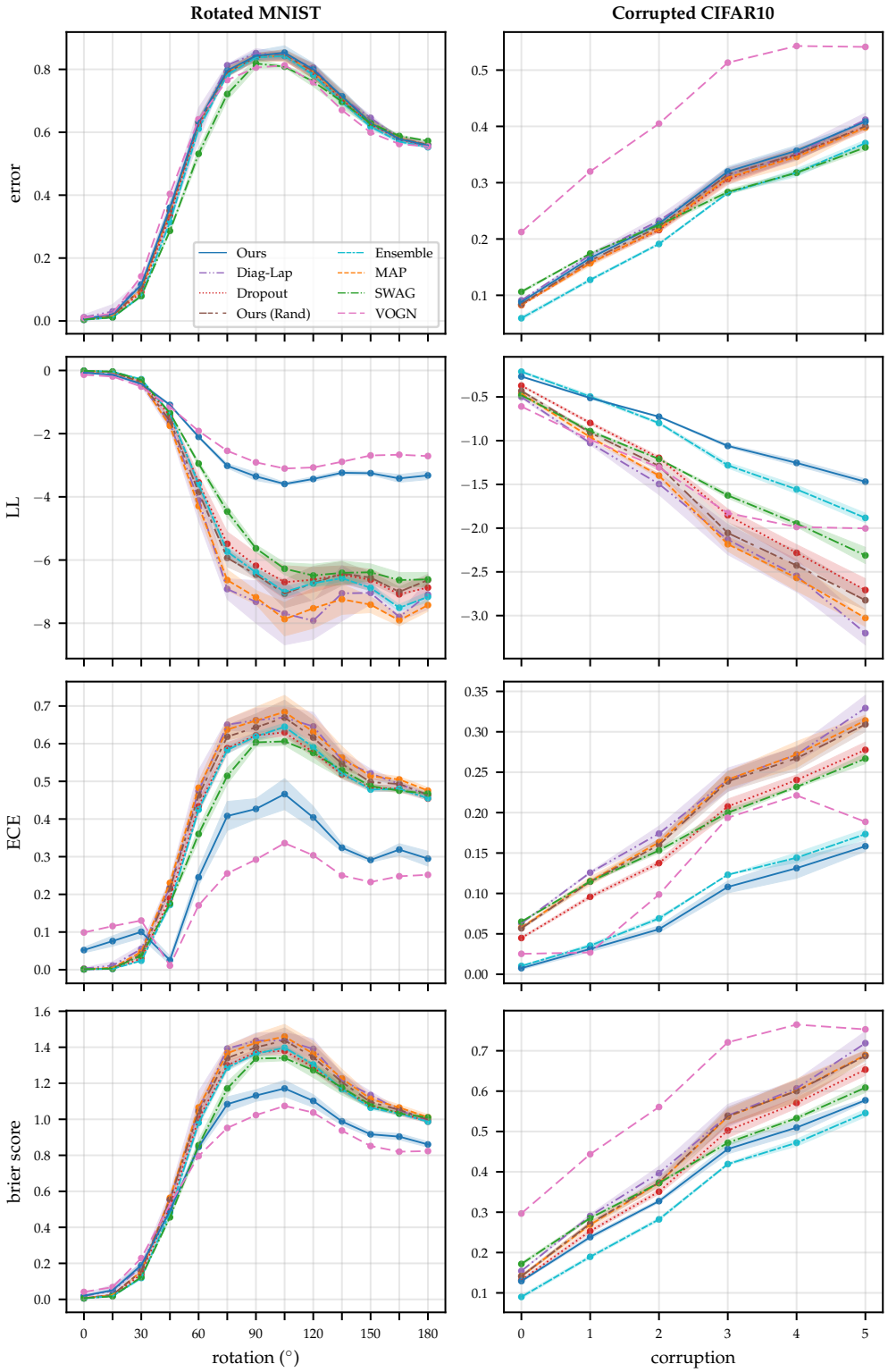

Figure 9. Full MNIST rotation and CIFAR10 corruption results, for ResNet-18, reporting predictive error, log-likelihood (LL), expected calibration error (ECE) and Brier score, respectively (from top to bottom).

For reference, we provide our results from Fig. 4 and Fig. 9 in numerical format in the tables below.

Table 2. MNIST – no rotation.

	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG	VOGN
LL	$-0.07_{\pm 0.01}$	$-0.01_{\pm 0.00}$	$-0.01_{\pm 0.00}$	$-0.04_{\pm 0.03}$	$-0.01_{\pm 0.00}$	$-0.01_{\pm 0.00}$	$-0.01_{\pm 0.00}$	$-0.14_{\pm nan}$
error	$0.01_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.01_{\pm 0.01}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.01_{\pm nan}$
ECE	$0.05_{\pm 0.01}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.10_{\pm nan}$
brier score	$0.02_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.02_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.04_{\pm nan}$

Table 3. MNIST – 15° rotation.

	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG	VOGN
LL	$-0.14_{\pm 0.02}$	$-0.05_{\pm 0.00}$	$-0.05_{\pm 0.00}$	$-0.11_{\pm 0.08}$	$-0.04_{\pm 0.00}$	$-0.05_{\pm 0.00}$	$-0.04_{\pm 0.00}$	$-0.19_{\pm nan}$
error	$0.02_{\pm 0.00}$	$0.02_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.03_{\pm 0.02}$	$0.01_{\pm 0.00}$	$0.02_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.02_{\pm nan}$
ECE	$0.08_{\pm 0.01}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.01_{\pm 0.01}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.12_{\pm nan}$
brier score	$0.05_{\pm 0.01}$	$0.03_{\pm 0.00}$	$0.02_{\pm 0.00}$	$0.05_{\pm 0.03}$	$0.02_{\pm 0.00}$	$0.02_{\pm 0.00}$	$0.02_{\pm 0.00}$	$0.07_{\pm nan}$

Table 4. MNIST – 30° rotation.

	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG	VOGN
LL	$-0.42_{\pm 0.04}$	$-0.36_{\pm 0.01}$	$-0.32_{\pm 0.02}$	$-0.44_{\pm 0.06}$	$-0.28_{\pm 0.02}$	$-0.39_{\pm 0.01}$	$-0.30_{\pm 0.00}$	$-0.51_{\pm nan}$
error	$0.11_{\pm 0.01}$	$0.10_{\pm 0.00}$	$0.09_{\pm 0.01}$	$0.12_{\pm 0.01}$	$0.08_{\pm 0.01}$	$0.10_{\pm 0.00}$	$0.08_{\pm 0.00}$	$0.14_{\pm nan}$
ECE	$0.10_{\pm 0.02}$	$0.04_{\pm 0.01}$	$0.03_{\pm 0.00}$	$0.06_{\pm 0.01}$	$0.02_{\pm 0.00}$	$0.05_{\pm 0.00}$	$0.04_{\pm 0.00}$	$0.13_{\pm nan}$
brier score	$0.19_{\pm 0.02}$	$0.16_{\pm 0.00}$	$0.14_{\pm 0.01}$	$0.18_{\pm 0.02}$	$0.12_{\pm 0.01}$	$0.16_{\pm 0.00}$	$0.12_{\pm 0.00}$	$0.23_{\pm nan}$

Table 5. MNIST – 45° rotation.

	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG	VOGN
LL	$-1.09_{\pm 0.03}$	$-1.60_{\pm 0.05}$	$-1.44_{\pm 0.11}$	$-1.68_{\pm 0.20}$	$-1.36_{\pm 0.07}$	$-1.75_{\pm 0.06}$	$-1.35_{\pm 0.02}$	$-1.15_{\pm nan}$
error	$0.36_{\pm 0.01}$	$0.35_{\pm 0.01}$	$0.33_{\pm 0.01}$	$0.35_{\pm 0.03}$	$0.31_{\pm 0.01}$	$0.35_{\pm 0.01}$	$0.29_{\pm 0.00}$	$0.40_{\pm nan}$
ECE	$0.03_{\pm 0.01}$	$0.22_{\pm 0.01}$	$0.19_{\pm 0.02}$	$0.22_{\pm 0.02}$	$0.17_{\pm 0.01}$	$0.23_{\pm 0.01}$	$0.18_{\pm 0.00}$	$0.01_{\pm nan}$
brier score	$0.49_{\pm 0.02}$	$0.55_{\pm 0.02}$	$0.52_{\pm 0.02}$	$0.55_{\pm 0.04}$	$0.48_{\pm 0.02}$	$0.56_{\pm 0.02}$	$0.46_{\pm 0.01}$	$0.53_{\pm nan}$

Table 6. MNIST – 60° rotation.

	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG	VOGN
LL	$-2.10_{\pm 0.03}$	$-3.85_{\pm 0.18}$	$-3.54_{\pm 0.23}$	$-4.11_{\pm 0.66}$	$-3.60_{\pm 0.10}$	$-4.29_{\pm 0.21}$	$-2.95_{\pm 0.08}$	$-1.92_{\pm nan}$
error	$0.63_{\pm 0.01}$	$0.63_{\pm 0.01}$	$0.62_{\pm 0.01}$	$0.62_{\pm 0.05}$	$0.61_{\pm 0.01}$	$0.63_{\pm 0.01}$	$0.53_{\pm 0.02}$	$0.64_{\pm nan}$
ECE	$0.25_{\pm 0.02}$	$0.46_{\pm 0.02}$	$0.43_{\pm 0.02}$	$0.47_{\pm 0.06}$	$0.42_{\pm 0.01}$	$0.48_{\pm 0.02}$	$0.36_{\pm 0.02}$	$0.17_{\pm nan}$
brier score	$0.85_{\pm 0.02}$	$1.04_{\pm 0.03}$	$1.00_{\pm 0.03}$	$1.05_{\pm 0.10}$	$0.98_{\pm 0.02}$	$1.07_{\pm 0.03}$	$0.86_{\pm 0.03}$	$0.80_{\pm nan}$

Table 7. MNIST – 75° rotation.

	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG	VOGN
LL	$-3.02_{\pm 0.07}$	$-5.93_{\pm 0.28}$	$-5.49_{\pm 0.38}$	$-6.92_{\pm 0.32}$	$-5.74_{\pm 0.15}$	$-6.63_{\pm 0.33}$	$-4.46_{\pm 0.18}$	$-2.54_{\pm nan}$
error	$0.80_{\pm 0.02}$	$0.79_{\pm 0.01}$	$0.79_{\pm 0.01}$	$0.81_{\pm 0.00}$	$0.78_{\pm 0.01}$	$0.79_{\pm 0.01}$	$0.72_{\pm 0.02}$	$0.77_{\pm nan}$
ECE	$0.41_{\pm 0.04}$	$0.62_{\pm 0.03}$	$0.59_{\pm 0.01}$	$0.65_{\pm 0.01}$	$0.58_{\pm 0.01}$	$0.64_{\pm 0.03}$	$0.51_{\pm 0.02}$	$0.26_{\pm nan}$
brier score	$1.08_{\pm 0.04}$	$1.34_{\pm 0.04}$	$1.30_{\pm 0.02}$	$1.39_{\pm 0.01}$	$1.29_{\pm 0.02}$	$1.37_{\pm 0.04}$	$1.17_{\pm 0.04}$	$0.95_{\pm nan}$

Table 8. MNIST – 90° rotation.

	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG	VOGN
LL	$-3.35_{\pm 0.13}$	$-6.46_{\pm 0.15}$	$-6.18_{\pm 0.41}$	$-7.32_{\pm 0.67}$	$-6.39_{\pm 0.17}$	$-7.18_{\pm 0.22}$	$-5.63_{\pm 0.12}$	$-2.91_{\pm nan}$
error	$0.84_{\pm 0.02}$	$0.84_{\pm 0.01}$	$0.84_{\pm 0.01}$	$0.85_{\pm 0.01}$	$0.84_{\pm 0.01}$	$0.84_{\pm 0.01}$	$0.82_{\pm 0.02}$	$0.81_{\pm nan}$
ECE	$0.43_{\pm 0.03}$	$0.64_{\pm 0.04}$	$0.62_{\pm 0.01}$	$0.66_{\pm 0.03}$	$0.62_{\pm 0.01}$	$0.66_{\pm 0.04}$	$0.60_{\pm 0.01}$	$0.29_{\pm nan}$
brier score	$1.13_{\pm 0.03}$	$1.40_{\pm 0.05}$	$1.37_{\pm 0.01}$	$1.44_{\pm 0.04}$	$1.36_{\pm 0.01}$	$1.43_{\pm 0.05}$	$1.34_{\pm 0.02}$	$1.02_{\pm nan}$

Table 9. MNIST – 105° rotation.

	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG	VOGN
LL	$-3.59_{\pm 0.05}$	$-7.06_{\pm 0.45}$	$-6.70_{\pm 0.52}$	$-7.69_{\pm 0.99}$	$-7.01_{\pm 0.17}$	$-7.87_{\pm 0.53}$	$-6.28_{\pm 0.19}$	$-3.10_{\pm nan}$
error	$0.85_{\pm 0.02}$	$0.84_{\pm 0.02}$	$0.84_{\pm 0.01}$	$0.85_{\pm 0.01}$	$0.84_{\pm 0.01}$	$0.84_{\pm 0.02}$	$0.81_{\pm 0.00}$	$0.81_{\pm nan}$
ECE	$0.47_{\pm 0.04}$	$0.67_{\pm 0.05}$	$0.63_{\pm 0.01}$	$0.67_{\pm 0.03}$	$0.64_{\pm 0.01}$	$0.68_{\pm 0.04}$	$0.61_{\pm 0.01}$	$0.34_{\pm nan}$
brier score	$1.17_{\pm 0.05}$	$1.44_{\pm 0.07}$	$1.38_{\pm 0.02}$	$1.44_{\pm 0.04}$	$1.40_{\pm 0.01}$	$1.46_{\pm 0.07}$	$1.34_{\pm 0.02}$	$1.07_{\pm nan}$

Table 10. MNIST – 120° rotation.

	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG	VOGN
LL	$-3.43_{\pm 0.07}$	$-6.73_{\pm 0.53}$	$-6.62_{\pm 0.39}$	$-7.92_{\pm 0.59}$	$-6.73_{\pm 0.11}$	$-7.53_{\pm 0.63}$	$-6.49_{\pm 0.36}$	$-3.07_{\pm nan}$
error	$0.80_{\pm 0.02}$	$0.79_{\pm 0.02}$	$0.78_{\pm 0.01}$	$0.81_{\pm 0.01}$	$0.78_{\pm 0.01}$	$0.79_{\pm 0.02}$	$0.76_{\pm 0.02}$	$0.76_{\pm nan}$
ECE	$0.40_{\pm 0.03}$	$0.62_{\pm 0.05}$	$0.58_{\pm 0.01}$	$0.65_{\pm 0.04}$	$0.59_{\pm 0.01}$	$0.63_{\pm 0.04}$	$0.58_{\pm 0.03}$	$0.30_{\pm nan}$
brier score	$1.10_{\pm 0.03}$	$1.35_{\pm 0.07}$	$1.29_{\pm 0.02}$	$1.39_{\pm 0.06}$	$1.30_{\pm 0.01}$	$1.36_{\pm 0.07}$	$1.27_{\pm 0.04}$	$1.04_{\pm nan}$

Table 11. MNIST – 135° rotation.

	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG	VOGN
LL	$-3.24_{\pm 0.06}$	$-6.43_{\pm 0.38}$	$-6.46_{\pm 0.28}$	$-7.05_{\pm 0.88}$	$-6.57_{\pm 0.10}$	$-7.24_{\pm 0.48}$	$-6.40_{\pm 0.37}$	$-2.89_{\pm nan}$
error	$0.71_{\pm 0.02}$	$0.71_{\pm 0.02}$	$0.70_{\pm 0.01}$	$0.71_{\pm 0.01}$	$0.70_{\pm 0.01}$	$0.71_{\pm 0.02}$	$0.70_{\pm 0.02}$	$0.67_{\pm nan}$
ECE	$0.32_{\pm 0.01}$	$0.55_{\pm 0.03}$	$0.52_{\pm 0.01}$	$0.56_{\pm 0.02}$	$0.52_{\pm 0.01}$	$0.56_{\pm 0.03}$	$0.53_{\pm 0.02}$	$0.25_{\pm nan}$
brier score	$0.99_{\pm 0.02}$	$1.21_{\pm 0.05}$	$1.17_{\pm 0.02}$	$1.22_{\pm 0.04}$	$1.17_{\pm 0.01}$	$1.23_{\pm 0.05}$	$1.18_{\pm 0.04}$	$0.94_{\pm nan}$

Table 12. MNIST – 150° rotation.

	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG	VOGN
LL	$-3.25_{\pm 0.05}$	$-6.56_{\pm 0.18}$	$-6.62_{\pm 0.33}$	$-7.04_{\pm 0.36}$	$-6.88_{\pm 0.11}$	$-7.41_{\pm 0.25}$	$-6.39_{\pm 0.27}$	$-2.69_{\pm nan}$
error	$0.63_{\pm 0.02}$	$0.63_{\pm 0.01}$	$0.63_{\pm 0.00}$	$0.65_{\pm 0.01}$	$0.62_{\pm 0.01}$	$0.63_{\pm 0.01}$	$0.63_{\pm 0.01}$	$0.60_{\pm nan}$
ECE	$0.29_{\pm 0.01}$	$0.50_{\pm 0.01}$	$0.48_{\pm 0.01}$	$0.52_{\pm 0.01}$	$0.48_{\pm 0.01}$	$0.51_{\pm 0.01}$	$0.49_{\pm 0.01}$	$0.23_{\pm nan}$
brier score	$0.92_{\pm 0.02}$	$1.10_{\pm 0.02}$	$1.07_{\pm 0.01}$	$1.13_{\pm 0.02}$	$1.06_{\pm 0.01}$	$1.11_{\pm 0.02}$	$1.08_{\pm 0.02}$	$0.85_{\pm nan}$

Table 13. MNIST – 165° rotation.

	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG	VOGN
LL	$-3.42_{\pm 0.12}$	$-7.01_{\pm 0.15}$	$-7.08_{\pm 0.39}$	$-7.80_{\pm 0.12}$	$-7.51_{\pm 0.11}$	$-7.91_{\pm 0.18}$	$-6.63_{\pm 0.24}$	$-2.67_{\pm nan}$
error	$0.58_{\pm 0.01}$	$0.58_{\pm 0.01}$	$0.58_{\pm 0.01}$	$0.58_{\pm 0.00}$	$0.57_{\pm 0.01}$	$0.58_{\pm 0.01}$	$0.59_{\pm 0.00}$	$0.56_{\pm nan}$
ECE	$0.32_{\pm 0.02}$	$0.49_{\pm 0.01}$	$0.48_{\pm 0.01}$	$0.49_{\pm 0.01}$	$0.48_{\pm 0.00}$	$0.51_{\pm 0.01}$	$0.48_{\pm 0.00}$	$0.25_{\pm nan}$
brier score	$0.90_{\pm 0.02}$	$1.05_{\pm 0.01}$	$1.04_{\pm 0.01}$	$1.05_{\pm 0.01}$	$1.03_{\pm 0.01}$	$1.07_{\pm 0.02}$	$1.03_{\pm 0.01}$	$0.82_{\pm nan}$

Table 14. MNIST – 180° rotation.

	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG	VOGN
LL	$-3.32_{\pm 0.13}$	$-6.63_{\pm 0.18}$	$-6.87_{\pm 0.32}$	$-7.10_{\pm 0.47}$	$-7.16_{\pm 0.16}$	$-7.43_{\pm 0.20}$	$-6.61_{\pm 0.22}$	$-2.71_{\pm nan}$
error	$0.56_{\pm 0.01}$	$0.56_{\pm 0.01}$	$0.56_{\pm 0.00}$	$0.55_{\pm 0.01}$	$0.55_{\pm 0.00}$	$0.56_{\pm 0.01}$	$0.57_{\pm 0.00}$	$0.55_{\pm nan}$
ECE	$0.29_{\pm 0.02}$	$0.46_{\pm 0.01}$	$0.45_{\pm 0.00}$	$0.46_{\pm 0.00}$	$0.46_{\pm 0.01}$	$0.48_{\pm 0.01}$	$0.47_{\pm 0.01}$	$0.25_{\pm nan}$
brier score	$0.86_{\pm 0.02}$	$1.00_{\pm 0.01}$	$0.99_{\pm 0.01}$	$0.99_{\pm 0.01}$	$0.99_{\pm 0.00}$	$1.01_{\pm 0.02}$	$1.01_{\pm 0.01}$	$0.82_{\pm nan}$

Table 15. CIFAR10 – no corruption.

	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG	VOGN
LL	$-0.27_{\pm 0.00}$	$-0.43_{\pm 0.01}$	$-0.37_{\pm 0.01}$	$-0.50_{\pm 0.02}$	$-0.21_{\pm 0.01}$	$-0.46_{\pm 0.02}$	$-0.48_{\pm 0.01}$	$-0.61_{\pm nan}$
error	$0.09_{\pm 0.00}$	$0.08_{\pm 0.00}$	$0.08_{\pm 0.00}$	$0.09_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.08_{\pm 0.00}$	$0.11_{\pm 0.00}$	$0.21_{\pm nan}$
ECE	$0.01_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.04_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.07_{\pm 0.00}$	$0.03_{\pm nan}$
brier score	$0.13_{\pm 0.00}$	$0.14_{\pm 0.00}$	$0.13_{\pm 0.00}$	$0.15_{\pm 0.00}$	$0.09_{\pm 0.00}$	$0.14_{\pm 0.00}$	$0.17_{\pm 0.00}$	$0.30_{\pm nan}$

Table 16. CIFAR10 – level 1 corruption.

	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG	VOGN
LL	$-0.51_{\pm 0.01}$	$-0.91_{\pm 0.01}$	$-0.80_{\pm 0.02}$	$-1.03_{\pm 0.02}$	$-0.50_{\pm 0.02}$	$-0.96_{\pm 0.02}$	$-0.89_{\pm 0.02}$	$-0.99_{\pm nan}$
error	$0.17_{\pm 0.01}$	$0.16_{\pm 0.00}$	$0.16_{\pm 0.00}$	$0.17_{\pm 0.00}$	$0.13_{\pm 0.00}$	$0.16_{\pm 0.00}$	$0.17_{\pm 0.00}$	$0.32_{\pm nan}$
ECE	$0.03_{\pm 0.00}$	$0.11_{\pm 0.00}$	$0.10_{\pm 0.00}$	$0.13_{\pm 0.00}$	$0.04_{\pm 0.00}$	$0.12_{\pm 0.01}$	$0.11_{\pm 0.00}$	$0.03_{\pm nan}$
brier score	$0.24_{\pm 0.00}$	$0.27_{\pm 0.00}$	$0.25_{\pm 0.00}$	$0.29_{\pm 0.00}$	$0.19_{\pm 0.00}$	$0.27_{\pm 0.01}$	$0.29_{\pm 0.00}$	$0.44_{\pm nan}$

Table 17. CIFAR10 – level 2 corruption.

	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG	VOGN
LL	$-0.73_{\pm 0.01}$	$-1.29_{\pm 0.06}$	$-1.20_{\pm 0.02}$	$-1.50_{\pm 0.12}$	$-0.80_{\pm 0.01}$	$-1.40_{\pm 0.03}$	$-1.21_{\pm 0.00}$	$-1.31_{\pm nan}$
error	$0.23_{\pm 0.00}$	$0.22_{\pm 0.01}$	$0.22_{\pm 0.00}$	$0.23_{\pm 0.01}$	$0.19_{\pm 0.00}$	$0.22_{\pm 0.00}$	$0.22_{\pm 0.00}$	$0.40_{\pm nan}$
ECE	$0.06_{\pm 0.00}$	$0.16_{\pm 0.01}$	$0.14_{\pm 0.00}$	$0.17_{\pm 0.01}$	$0.07_{\pm 0.00}$	$0.16_{\pm 0.00}$	$0.15_{\pm 0.00}$	$0.10_{\pm nan}$
brier score	$0.33_{\pm 0.00}$	$0.37_{\pm 0.01}$	$0.35_{\pm 0.01}$	$0.40_{\pm 0.02}$	$0.28_{\pm 0.00}$	$0.37_{\pm 0.01}$	$0.37_{\pm 0.00}$	$0.56_{\pm nan}$

Table 18. CIFAR10 – level 3 corruption.

	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG	VOGN
LL	$-1.06_{\pm 0.02}$	$-2.06_{\pm 0.12}$	$-1.85_{\pm 0.07}$	$-2.13_{\pm 0.17}$	$-1.28_{\pm 0.03}$	$-2.18_{\pm 0.08}$	$-1.63_{\pm 0.03}$	$-1.83_{\pm nan}$
error	$0.32_{\pm 0.01}$	$0.31_{\pm 0.01}$	$0.31_{\pm 0.01}$	$0.31_{\pm 0.01}$	$0.28_{\pm 0.00}$	$0.31_{\pm 0.01}$	$0.28_{\pm 0.00}$	$0.51_{\pm nan}$
ECE	$0.11_{\pm 0.01}$	$0.24_{\pm 0.01}$	$0.21_{\pm 0.01}$	$0.24_{\pm 0.01}$	$0.12_{\pm 0.00}$	$0.24_{\pm 0.01}$	$0.20_{\pm 0.00}$	$0.19_{\pm nan}$
brier score	$0.46_{\pm 0.01}$	$0.54_{\pm 0.02}$	$0.50_{\pm 0.02}$	$0.54_{\pm 0.03}$	$0.42_{\pm 0.00}$	$0.54_{\pm 0.02}$	$0.47_{\pm 0.01}$	$0.72_{\pm nan}$

Table 19. CIFAR10 – level 4 corruption.

	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG	VOGN
LL	$-1.25_{\pm 0.03}$	$-2.43_{\pm 0.18}$	$-2.28_{\pm 0.10}$	$-2.54_{\pm 0.18}$	$-1.56_{\pm 0.05}$	$-2.57_{\pm 0.15}$	$-1.95_{\pm 0.04}$	$-1.99_{\pm nan}$
error	$0.36_{\pm 0.01}$	$0.35_{\pm 0.01}$	$0.35_{\pm 0.01}$	$0.35_{\pm 0.01}$	$0.32_{\pm 0.01}$	$0.35_{\pm 0.01}$	$0.32_{\pm 0.00}$	$0.54_{\pm nan}$
ECE	$0.13_{\pm 0.01}$	$0.27_{\pm 0.01}$	$0.24_{\pm 0.01}$	$0.27_{\pm 0.01}$	$0.14_{\pm 0.01}$	$0.27_{\pm 0.02}$	$0.23_{\pm 0.00}$	$0.22_{\pm nan}$
brier score	$0.51_{\pm 0.02}$	$0.60_{\pm 0.03}$	$0.57_{\pm 0.01}$	$0.61_{\pm 0.02}$	$0.47_{\pm 0.01}$	$0.60_{\pm 0.03}$	$0.53_{\pm 0.00}$	$0.76_{\pm nan}$

Table 20. CIFAR10 – level 5 corruption.

	OURS	OURS (RAND)	DROPOUT	DIAG-LAP	ENSEMBLE	MAP	SWAG	VOGN
LL	$-1.47_{\pm 0.03}$	$-2.82_{\pm 0.11}$	$-2.71_{\pm 0.13}$	$-3.20_{\pm 0.13}$	$-1.88_{\pm 0.05}$	$-3.03_{\pm 0.10}$	$-2.31_{\pm 0.09}$	$-2.00_{\pm nan}$
error	$0.41_{\pm 0.00}$	$0.40_{\pm 0.01}$	$0.40_{\pm 0.01}$	$0.41_{\pm 0.01}$	$0.37_{\pm 0.01}$	$0.40_{\pm 0.00}$	$0.36_{\pm 0.01}$	$0.54_{\pm nan}$
ECE	$0.16_{\pm 0.01}$	$0.31_{\pm 0.01}$	$0.28_{\pm 0.01}$	$0.33_{\pm 0.02}$	$0.17_{\pm 0.01}$	$0.31_{\pm 0.01}$	$0.27_{\pm 0.01}$	$0.19_{\pm nan}$
brier score	$0.58_{\pm 0.00}$	$0.69_{\pm 0.01}$	$0.65_{\pm 0.01}$	$0.72_{\pm 0.03}$	$0.55_{\pm 0.01}$	$0.69_{\pm 0.01}$	$0.61_{\pm 0.01}$	$0.75_{\pm nan}$

B. Derivations for the Wasserstein Pruning Objective

B.1. Derivation for (22)

Note that, for our linearized model (described in Section 3), the true posterior $p(\mathbf{w}|\mathcal{D})$ is either Gaussian or approximately Gaussian. Additionally, the approximate posterior $q_S(\mathbf{w}) = q(\mathbf{w}_S) \prod_r \delta(\mathrm{w}_r - \widehat{\mathrm{w}}_r)$ can be seen as a degenerate Gaussian in which rows and columns of the covariance matrix are zeroed out. Thus, we consider the squared 2-Wasserstein distance between two Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, which has the following closed-form expression (Givens et al., 1984)[3]:

$$W_2\left(\mathcal{N}\left(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1\right), \mathcal{N}\left(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2\right)\right)^2 = \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2^2 + \mathrm{Tr}\left(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2 - 2\left(\boldsymbol{\Sigma}_2^{1/2}\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_2^{1/2}\right)^{1/2}\right) . \tag{24}$$

In this case both distributions have the same mean: $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \widehat{\mathbf{w}}$. The true posterior's covariance matrix is the inverse GGN matrix, i.e. $\boldsymbol{\Sigma}_1 = \widetilde{\mathbf{H}}^{-1}$. For the approximate posterior $\boldsymbol{\Sigma}_2 = \widetilde{\mathbf{H}}_{S+}^{-1}$, which is equal to $\widetilde{\mathbf{H}}_S^{-1}$ (the inverse GGN matrix of the subnetwork) padded with zeros at the positions corresponding to point estimated weights w_r, matching the shape of $\widetilde{\mathbf{H}}^{-1}$. Alternatively, but equivalently, we can define $\widetilde{\mathbf{H}}_{S+}^{-1} = \mathbf{M}_S \odot \widetilde{\mathbf{H}}^{-1}$, where $\odot$ is the Hadamard product, and $\mathbf{M}_S$ is a mask matrix with zeros in the rows and columns corresponding to w_r, i.e. the rows and columns corresponding to weights not included in the subnetwork. This gives us:

$$\begin{aligned}
&W_2(p(\mathbf{w}|\mathcal{D}), q_S(\mathbf{w}))^2 \\
&= W_2\left(\mathcal{N}(\widehat{\mathbf{w}}, \widetilde{\mathbf{H}}^{-1}), \mathcal{N}(\widehat{\mathbf{w}}, \widetilde{\mathbf{H}}_{S+}^{-1})\right)^2 \\
&= \cancel{\|\widehat{\mathbf{w}} - \widehat{\mathbf{w}}\|_2^2} + \mathrm{Tr}\left(\widetilde{\mathbf{H}}^{-1} + \widetilde{\mathbf{H}}_{S+}^{-1} - 2\left(\widetilde{\mathbf{H}}_{S+}^{-1/2}\widetilde{\mathbf{H}}^{-1}\widetilde{\mathbf{H}}_{S+}^{-1/2}\right)^{1/2}\right) \\
&= \mathrm{Tr}\left(\widetilde{\mathbf{H}}^{-1} + \widetilde{\mathbf{H}}_{S+}^{-1} - 2\left(\widetilde{\mathbf{H}}_{S+}^{-1/2}\widetilde{\mathbf{H}}^{-1}\widetilde{\mathbf{H}}_{S+}^{-1/2}\right)^{1/2}\right) .
\end{aligned}$$

B.2. Derivation for (23)

For $\widetilde{\mathbf{H}}^{-1} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_D^2)$, the Wasserstein pruning objective in (22) simplifies to

$$\begin{aligned}
&W_2(p(\mathbf{w}|\mathcal{D}), q_S(\mathbf{w}))^2 \\
&= \mathrm{Tr}\left(\widetilde{\mathbf{H}}^{-1}\right) + \mathrm{Tr}\left(\widetilde{\mathbf{H}}_{S+}^{-1}\right) - 2\mathrm{Tr}\left(\widetilde{\mathbf{H}}^{-1/2}\widetilde{\mathbf{H}}_{S+}^{-1/2}\right) \\
&= \sum_{d=1}^{D} \sigma_d^2 + m_d\sigma_d^2 - 2m_d\sigma_d^2 \\
&= \sum_{d=1}^{D} \sigma_d^2(1 - m_d) ,
\end{aligned}$$

where m_d is the d^{th} diagonal element of $\mathbf{M}_S$, i.e. $m_d = 1$ if w_d is included in the subnetwork or 0 otherwise.

C. Updating the prior precision for uncertainty estimation with subnetworks

As described in Section 3, the linearised Laplace method can be understood as approximating our NN with a basis function linear model, where the jacobian of the NN evaluated at $\mathbf{x}$, $\mathbf{J}(\mathbf{x}) \in \mathcal{R}^{O \times D}$ represents the feature expansion. When employing an Isotropic Gaussian prior with precision λ and for a given output dimension i, this formulation corresponds to a Gaussian process with kernel

$$k_i(\mathbf{x}, \mathbf{x}') = \lambda^{-1}\mathbf{J}(\mathbf{x})_i\mathbf{J}(\mathbf{x}')_i^\top = \lambda^{-1}\sum_{d=1}^{D}\mathbf{J}(\mathbf{x})_{i,d}\mathbf{J}(\mathbf{x}')_{i,d}. \tag{25}$$

[3] This also holds for our case of a degenerate Gaussian with singular covariance matrix (Givens et al., 1984).

For our subnetwork model, the Jacobian feature expansion is $\mathbf{J}_S(\mathbf{x}) \in \mathcal{R}^{O \times S}$, which is a submatrix of $\mathbf{J}(\mathbf{x})$. It follows that the implied kernel will be computed in the same way as (25), removing $D - S$ terms from the sum. The updated prior precision $\lambda_S = \lambda \cdot {}^{S}/_{D}$ aims to maintain the magnitude of the sum, thus making the kernel corresponding to the subnetwork as similar as possible to that of the full network.

D. Experimental Setup

D.1. Toy Experiments

We train a single, 2 hidden layer network, with 50 hidden ReLU units per layer using MAP inference until convergence. Specifically, we use SGD with a learning rate of 1×10^{-3}, momentum of 0.9 and weight decay of 1×10^{-4}. We use a batch size of 512. The objective we optimise is the Gaussian log-likelihood of our data, where the mean is outputted by the network and the the variance is a hyperparameter learnt jointly with NN parameters by SGD. This variance parameters is shared among all datapoints. Once the network is trained, we perform post-hoc inference on it using different approaches. Since all of these involve the linearized approximation, the mean prediction is the same for all methods. Only their uncertainty estimates vary.

Note that while for this toy example, we could in principle use the full covariance matrix for the purpose of subnetwork selection, we still just use its diagonal (as described in Section 5) for consistency. We use GGN Laplace inference over network weights (not biases) in combination with the linearized predictive distribution in (12). Thus, all approaches considered share their predictive mean, allowing us to better compare their uncertainty estimates.

All approaches share a single prior precision of $\lambda = 3$, scaled as $\lambda_S = \lambda \cdot {}^{S}/_{D}$. We choose this prior precision such that the full covariance approach (optimistic baseline), where $\lambda_S = \lambda$, presents reasonable results. We first tried a precision of 1 and found the full covariance approach to produce excessively large errorbars (covering the whole plot). A value of 3 produces more reasonable results.

Final layer inference is performed by computing the full Laplace covariance matrix and discarding all entries except those corresponding to the final layer of the NN. Results for random sub-network selection are obtained with a single sample from a scaled uniform distribution over weight choice.

D.2. UCI Experiments

In this experiment, our fully connected NNs have numbers of hidden layers $h_d{=}\{1, 2\}$ and hidden layer widths $w_d{=}\{50, 100\}$. For a dataset with input dimension i_d, the number of weights is given by $D{=}(i_d{+}1)w_d{+}(h_d{-}1)w_d^2$. Our 2 hidden layer, 100 hidden unit models have a weight count of the order 10^4. The non-linearity used is ReLU.

We first obtain a MAP estimate of each model's weights. Specifically, we use SGD with a learning rate of 1×10^{-3}, momentum of 0.9 and weight decay of 1×10^{-4}. We use a batch size of 512. The objective we optimise is the Gaussian log-likelihood of our data, where the mean is outputted by the network and the the variance is a hyperparameter learnt jointly with NN parameters by SGD.

For each dataset split, we set aside 15% of the train data as a validation set. We use these for early stopping training. Training runs for a maximum of 2000 epochs but early stops with a patience of 500 if validation performance does not increase. For the larger Protein dataset, these values are 500 and 125. The weight settings which provide best validation performance are kept.

We then perform full network GGN Laplace inference for each model. We also use our proposed Wassertein rule together with the diagonal Hessian assumption to prune every network's weight variances such that the number of variances that remain matches the size of every smaller network under consideration. The prior precision used for these steps is chosen such that the resulting predictor's logliklihood performance on the validation set is maximised. Specifically, we employ a grid search over the values: $\lambda : [0.0001, 0.001, 0.1, 0.5, 1, 2, 5, 10, 100, 1000]$. In all cases, we employ the linearized predictive in (12). Consequently, networks with the same number of weights make the same mean predictions. Increasing the number of weight variances considered will thus only increase predictive uncertainty.

D.3. Image Experiments

The results shown in Section 6.3 and App. A are obtained by training ResNet-18 (and ResNet-50) models using SGD with momentum. For each experiment repetition, we train 7 different models: The first is for: 'MAP', 'Ours', 'Ours (Rand)', 'SWAG', 'Diag-Laplace' and as the first element of 'Ensemble'. We train 4 additional 'Ensemble' elements, 1 network with 'Dropout', and, finally 1 network for 'VOGN'. The methods 'Ours', 'Ours (Rand)', 'SWAG', and 'Diag-Laplace' are applied post training.

For all methods except 'VOGN' we use the following training procedure. The (initial) learning rate, momentum, and weight decay are 0.1, 0.9, and 1×10^{-4}, respectively. For 'MAP' we use 4 Nvidia P100 GPUs with a total batch size of 2048. For the calculation of the Jacobian in the subnetwork selection phase we use a single P100 GPU with a batch size of 4. For the calculation of the hessian we use a single P100 GPU with a batch size of 2. We train on 1 Nvidia P100 GPU with a batch size of 256 for all other methods. Each dataset is trained for a different number of epochs, shown in Table 21. We decay the learning rate by a factor of 10 at scheduled epochs, also shown in Table 21. Otherwise, all methods and datasets share hyperparameters. These hyperparameter settings are the defaults provided by `PyTorch` for training on ImageNet. We found them to perform well across the board. We report results obtained at the final training epoch. We do not use a separate validation set to determine the best epoch as we found ResNet-18 and ResNet-50 to not overfit with the chosen schedules.

Table 21. Per-dataset training configuration for image experiments.

DATASET	NO. EPOCHS	LR SCHEDULE
MNIST	90	40, 70
CIFAR10	300	150, 225

For 'Dropout', we add dropout to the standard ResNet-50 model (He et al., 2016) in between the 2nd and 3rd convolutions in the bottleneck blocks. This approach follows Zagoruyko & Komodakis (2016) and Ashukha et al. (2020) who add dropout in-between the two convolutions of a WideResNet-50's basic block. Following Antorán et al. (2020), we choose a dropout probability of 0.1, as they found it to perform better than the value of 0.3 suggested by Ashukha et al. (2020). We use 16 MC samples for predictions. 'Ensemble' uses 5 elements for prediction. Ensemble elements differ from each other in their initialisation, which is sampled from the He initialisation distribution (He et al., 2015). We do not use adversarial training as, inline with Ashukha et al. (2020), we do not find it to improve results. For 'VOGN' we use the same procedure and hyperparameters as used by Osawa et al. (2019) in their CIFAR10 experiments, with the exception that we use a learning rate of 1×10^{-3} as we we found a value of 1×10^{-4} not to result in convergence. We train on a single Nvidia P100 GPU with a batch size of 256. See the authors' GitHub for more details: `github.com/team-approx-bayes/dl-with-bayes/blob/master/distributed/classification/configs/cifar10/resnet18_vogn_bs256_8gpu.json`.

We modify the standard ResNet-50 and ResNet-18 architectures such that the first 7×7 convolution is replaced with a 3×3 convolution. Additionally, we remove the first max-pooling layer. Following (Goyal et al., 2017), we zero-initialise the last batch normalisation layer in residual blocks so that they act as identity functions at the start of training.

At test time, we tune the prior precision used for 'Ours', 'Diag-Laplace' and 'SWAG' approximation on a validation set for each approach individually, as in Ritter et al. (2018); Kristiadi et al. (2020). We use a grid search from 1×10^{-4} to 1×10^{4} in logarithmic steps, and then a second, finer-grained grid search between the two best performing values (again with logarithmic steps).

D.4. Datasets

The 1d toy dataset used in Section 6.1 was taken from (Antorán et al., 2020). We obtained it from the authors' github repo: `https://github.com/cambridge-mlg/DUN`. Table 22 summarises the datasets used in Section 6.2.

We employ the Wine, Kin8nm and Protein datasets, together with their gap variants, because we find our models' performance to be most dependent on the quality of the estimated uncertainty here. On most other commonly used UCI regression datasets (Hernández-Lobato & Adams, 2015) we find increased uncertainty to hurt LL performance. In other words, the predictions made when using the MAP setting of the weights are better than those from any Bayesian ensemble.

Wine and Protein are available from the UCI dataset repository (Dua & Graff, 2017). Kin8nm is available from `https://www.openml.org/d/189` (Foong et al., 2019b). For the standard splits (Hernández-Lobato & Adams, 2015) 90% of

the data is used for training and 10% for validation. For the gap splits (Foong et al., 2019b) a split is obtained per input dimension by ordering points by their values across that dimension and removing the middle 33% of the points. These are used for validation.

The datasets used for our image experiments are outlined in Table 23.

Table 22. Datasets from tabular regression used in Section 6.2

Dataset	N Train	N Val (15% train)	N Test	Splits	Output Dim	Output Type	Input Dim	Input Type
Wine	1223	216	160	20	1	Continous	11	Continous
Wine Gap	906	161	532	11	1	Continous	11	Continous
Kin8nm	6267	1106	819	20	1	Continous	8	Continous
Kin8nm Gap	4642	820	2730	8	1	Continous	8	Continous
Protein	34983	6174	4573	5	1	Continous	9	Continous
Protein Gap	25913	4573	15244	9	1	Continous	9	Continous

Table 23. Summary of image datasets. The test and train set sizes are shown in brackets, e.g. (test & train).

NAME	SIZE	INPUT DIM.	NO. CLASSES	NO. SPLITS
MNIST (LeCun et al., 1998)	70,000 (60,000 & 10,000)	784 (28×28)	10	2
Fashion-MNIST (Xiao et al., 2017)	70,000 (60,000 & 10,000)	784 (28×28)	10	2
CIFAR10 (Krizhevsky & Hinton, 2009)	60,000 (50,000 & 10,000)	3072 ($32 \times 32 \times 3$)	10	2
SVHN (Netzer et al., 2011)	99,289 (73,257 & 26,032)	3072 ($32 \times 32 \times 3$)	10	2