Supplementary Material for the Paper: Calibrating Deep Convolutional Gaussian Processes

G.-L. Tran[1], E. V. Bonilla[2], J. P. Cunningham[3], P. Michiardi[1] and M. Filippone[1]

[1]EURECOM, France
[2]CSIRO's Data61 and UNSW
[3]Columbia University, USA

1 Random Feature Expansion of the RBF Covariance

We report here the expansion of the popular Radial Basis Function (RBF) covariance. Following the convolutional representation of images in our CNN+GP(RF) model, the RBF covariance is defined as:

$$k_{\text{rbf}}(\mathbf{x}_i, \mathbf{x}_j|\boldsymbol{\Psi}, \boldsymbol{\theta}) = \sigma^2 \exp\left[-\left(\mathbf{c}(\mathbf{x}_i|\boldsymbol{\Psi}) - \mathbf{c}(\mathbf{x}_j|\boldsymbol{\Psi})\right)^\top \boldsymbol{\Lambda}^{-1} \left(\mathbf{c}(\mathbf{x}_i|\boldsymbol{\Psi}) - \mathbf{c}(\mathbf{x}_j|\boldsymbol{\Psi})\right)\right], \tag{1}$$

with $\boldsymbol{\theta} = (\sigma, \boldsymbol{\Lambda} = \text{diag}(\ell_1^2, \ldots, \ell_d^2))$. It is possible to express this covariance function as the Fourier transform of a non-negative measure $p(\boldsymbol{\omega})$ Rahimi and Recht (2008), where $\boldsymbol{\omega}$ are the so-called spectral frequencies. It is straightforward to verify that $p(\boldsymbol{\omega}) = \mathcal{N}\left(\boldsymbol{\omega}|\mathbf{0}, \boldsymbol{\Lambda}^{-1}\right)$. Stacking N_{RF} Monte Carlo samples from $p(\boldsymbol{\omega})$ into $\boldsymbol{\Omega}$ by column, we obtain

$$\boldsymbol{\Phi}_{\text{rbf}} = \sqrt{\frac{\sigma^2}{N_{\text{RF}}}} \left[\cos\left(\mathbf{C}(\mathbf{X}|\boldsymbol{\Psi})\,\boldsymbol{\Omega}\right), \sin\left(\mathbf{C}(\mathbf{X}|\boldsymbol{\Psi})\,\boldsymbol{\Omega}\right)\right], \tag{2}$$

where $\mathbf{C}(\mathbf{X}|\boldsymbol{\Psi})$ denotes the matrix resulting from the application of convolutional layers to the image training set $\mathbf{X}$, and the sin and cos functions are applied elementwise to their argument.

2 Variational Inference for the Proposed Model

2.1 CNN+GP(RF)

In CNN+GP(RF), the variational parameters we would like to optimize are $\mathbf{M}_w, \mathbf{M}_\psi$ and $\mathbf{M}_\Omega$. Our model parameters $\mathbf{W}, \boldsymbol{\Psi}$ and $\boldsymbol{\Omega}$ share an identical form for the approximate posterior and prior. Focusing on $\mathbf{W}$, its elements have a standard normal prior, and we assume that the posterior $q(\mathbf{W})$ is a mixture of two Gaussian distribution, which can be factorized over rows, governed by variational parameters $\mathbf{M}_w$:

$$q(\mathbf{W}) = \prod_{r=1}^{R} q(\mathbf{W}_r), \quad \text{with} \quad q(\mathbf{W}_r) = \pi_w \mathcal{N}\left(\mathbf{M}_{w_r}, \sigma^2 \mathbf{I}_D\right) + (1 - \pi_w)\mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbf{I}_D\right), \tag{3}$$

where $\pi_w \in [0, 1]$, $\sigma^2 \approx 0$ and $\mathbf{M}_{w_r} \in \mathbb{R}^D$. This form of posterior leads to the sampling procedure which characterizes dropout Gal and Ghahramani (2016a,b). Given the choice of $\sigma^2 \approx 0$, $\mathbf{W}$ can be sampled by introducing Bernoulli variables

$$\mathbf{W} = \mathbf{M}_w \,\text{diag}[\mathbf{z}_w] \qquad \text{with} \qquad (\mathbf{z}_w)_i \sim \text{Bernoulli}(\pi_w), \tag{4}$$

and similarly for $\boldsymbol{\Psi}$ and $\boldsymbol{\Omega}$.

All variational parameters are optimized to maximize the lower bound of marginal likelihood which is defined as follows

$$\log\left[p(\mathbf{Y}|\mathbf{X},\boldsymbol{\theta}\right] \geq \mathrm{E}_{q(\mathbf{W},\boldsymbol{\Psi},\boldsymbol{\Omega})}\left(\log\left[p\left(\mathbf{Y}|\mathbf{X},\mathbf{W},\boldsymbol{\Psi},\boldsymbol{\Omega},\boldsymbol{\theta}\right)\right]\right) - \mathrm{KL}\left[q\left(\mathbf{W},\boldsymbol{\Psi},\boldsymbol{\Omega}\right)\|p\left(\mathbf{W},\boldsymbol{\Psi},\boldsymbol{\Omega}|\boldsymbol{\theta}\right)\right] \tag{5}$$

The expectation in 5 can be unbiasedly estimated using Monte Carlo and also considering a mini-batch of size m

$$\mathrm{E}_{q(\mathbf{W},\boldsymbol{\Psi},\boldsymbol{\Omega})}\left(\log\left[p\left(\mathbf{Y}|\mathbf{X},\mathbf{W},\boldsymbol{\Psi},\boldsymbol{\Omega},\boldsymbol{\theta}\right)\right]\right) \approx \frac{n}{m}\frac{1}{N_{\mathrm{MC}}}\sum_{i=1}^{N_{\mathrm{MC}}}\sum_{k\in\mathcal{I}_m}\log\left[p\left(\mathbf{y}_k|\mathbf{x}_k,\mathbf{W}^{(i)},\boldsymbol{\Psi}^{(i)},\boldsymbol{\Omega}^{(i)},\boldsymbol{\theta}\right)\right], \tag{6}$$

where $\mathbf{W}^{(i)},\boldsymbol{\Psi}^{(i)},\boldsymbol{\Omega}^{(i)}$ is a sample from $q(\mathbf{W},\boldsymbol{\Psi},\boldsymbol{\Omega})$, and can be obtained via 4. $\mathcal{I}_m$ is a set of m indices to select a mini-batch of training points. In classification, each individual $p\left(\mathbf{y}_k|\mathbf{x}_k,\mathbf{W}^{(i)},\boldsymbol{\Psi}^{(i)},\boldsymbol{\Omega}^{(i)},\boldsymbol{\theta}\right)$ can be computed using a softmax transformation. The KL term can be approximated following Gal and Ghahramani (2016a), noting that the fact that we are treating $\boldsymbol{\Omega}$ variationally, gives rise to extra terms that involve the GP length-scale ℓ:

$$\mathrm{KL}\left[q\left(\mathbf{W},\boldsymbol{\Psi},\boldsymbol{\Omega}\right)\|p\left(\mathbf{W},\boldsymbol{\Psi},\boldsymbol{\Omega}|\boldsymbol{\theta}\right)\right] \approx \frac{\pi_w}{2}\left\|\mathbf{M}_w\right\|^2 + \frac{\pi_\psi}{2}\left\|\mathbf{M}_\psi\right\|^2 + \frac{\ell^2\pi_\Omega}{2}\left\|\mathbf{M}_\Omega\right\|^2 + N_{\mathrm{RF}}\,d\,\log\left(\ell^{-2}\right) \tag{7}$$

2.2 CNN+GP(SORF)

In CNN+GP(SORF), our proposed variational inference scheme is similar to the one in CNN+GP(RF), except that $\boldsymbol{\Omega}$ is replaced by $l^{-1}\sqrt{N_{\mathrm{RF}}}\mathbf{H}\mathbf{D}_1\mathbf{H}\mathbf{D}_2\mathbf{H}\mathbf{D}_3$, with length-scale l and $\mathbf{D}_i = \mathrm{diag}\left(\mathbf{d}_i\right)$ and $\mathbf{H}$ is the normalized Walsh-Hadamard matrix. Because $\mathbf{d}_i$ is Rademacher distributed, the form of prior and posterior in MCD proposed by Gal and Ghahramani (2016a,b) is inadequate. Therefore, we use the prior $p_\epsilon\left(\mathbf{d}_i\right) = \mathcal{N}\left(\mathbf{d}_i|\mathbf{d}_i^*,\epsilon^2\mathbf{I}_{N_{\mathrm{RF}}}\right)$ with $\mathbf{d}_i^*$ sampled from the Rademacher distribution and a small positive ϵ. The posterior $q\left(\mathbf{d}\right)$ is also composed by two Gaussian distribution as in CNN+GP(RF)

$$q\left(\mathbf{d}_i\right) = \prod_{j=1}^{N_{\mathrm{RF}}} q\left(\left[\mathbf{d}_i\right]_j\right), \text{ where } q\left(\left[\mathbf{d}_i\right]_j\right) = \pi_d\mathcal{N}\left(\mathbf{M}_{\left[\mathbf{d}_i\right]_j},\sigma^2\right) + \left(1-\pi_d\right)\mathcal{N}\left(\left[\mathbf{d}_i^*\right]_j,\sigma^2\right) \tag{8}$$

with $\pi_d \in [0,1], \sigma^2 \approx 0$ and $\mathbf{M}_{\mathbf{d}_i} \in \mathbb{R}^{N_{\mathrm{RF}}}$. Following Gal and Ghahramani (2016a), we can approximate the KL term between $q\left(\mathbf{d}_i\right)$ and $p\left(\mathbf{d}_i\right)$

$$\mathrm{KL}\left(q\left(\mathbf{d}_i\right)\|p_\epsilon\left(\mathbf{d}_i\right)\right) \approx \frac{\pi_d}{2\epsilon^2}\|\mathbf{M}_{\mathbf{d}_i} - \mathbf{d}_i^*\|^2 \tag{9}$$

In terms of implementation, we do not apply MCD to $\mathbf{d}_i - \mathbf{d}_i^*$ but on $\mathbf{d}_i$ directly. According to this choice, each element in $\mathbf{d}_i$ is sampled based on the variational parameters $M_{\mathbf{d}_i-\mathbf{d}_i^*}$ as in 10. Thanks to this trick, the implementation of MCD scheme does not change for optimizing $\mathbf{d}_i$

$$\mathbf{d}_i = \begin{cases}\mathbf{M}_{\mathbf{d}_i-\mathbf{d}_i^*} + \mathbf{d}_i^*, & \text{with probability } \pi_d\\ \mathbf{d}_i^*, & \text{otherwise}\end{cases} \tag{10}$$

In figure 1, we report some experimental results to illustrate the impact of optimizing $\mathbf{d}_i$. For CIFAR10-LeNet and CIFAR100-RESNET, the optimization of SORF parameters outperforms the case where spectral frequencies are fixed in terms of ERR, MNLL and BRIER. In the case of CIFAR10-RESNET, the gains are marginal.

2.3 Optimization for covariance parameters

When using 7 to optimize all variational parameters pertaining to $q\left(\mathbf{W},\boldsymbol{\Psi},\boldsymbol{\Omega}\right)$ jointly with covariance $\boldsymbol{\theta}$ we encountered some instabilities, and therefore we decided to report results when fixing the covariance

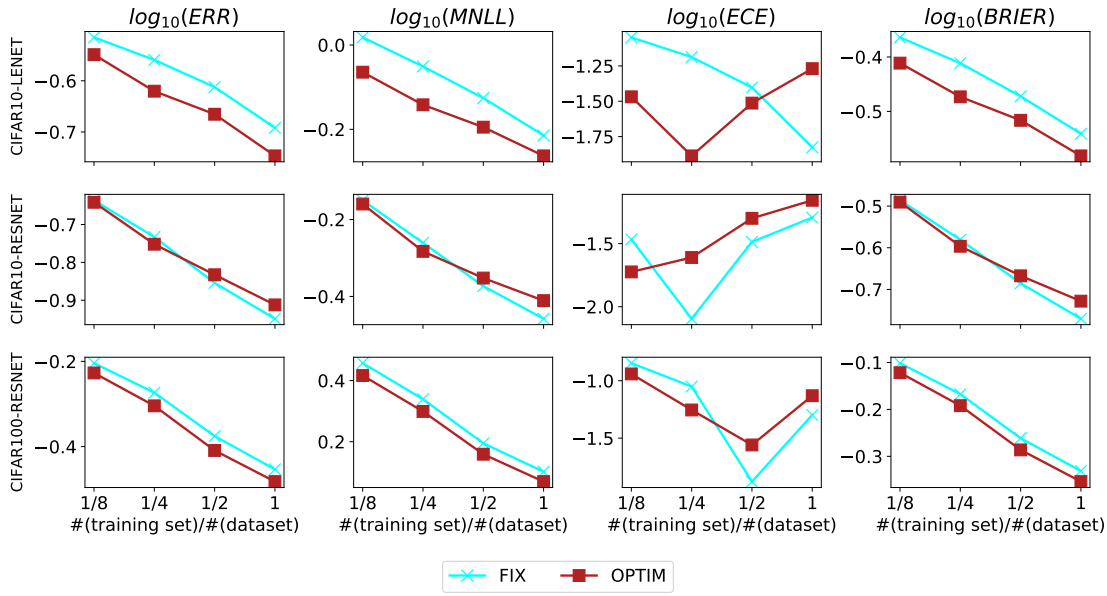

Figure 1: Impact of optimization of SORF parameters

parameters $\boldsymbol{\theta}$ in our paper. For the case where $\boldsymbol{\Omega}$ is not learned variationally we can simply draw $\boldsymbol{\Omega}$ from the prior $\mathcal{N}(\boldsymbol{\Omega}_{\cdot j}|\mathbf{0}, \Lambda^{-1})$ and consider the reparameterization:

$$\boldsymbol{\Omega}_{\cdot j} = \Lambda^{-\frac{1}{2}}\boldsymbol{\epsilon}, \tag{11}$$

where $\varepsilon_i \sim \mathcal{N}(\varepsilon_i|0,1)$ (Lázaro-Gredilla et al., 2010). This reparameterization allows for the update of covariance parameters $\boldsymbol{\theta}$ fixing the randomness in the sampling from $p(\boldsymbol{\Omega}|\boldsymbol{\theta})$. The results comparing CNN+GP(SORF) when updating or fixing $\boldsymbol{\theta}$ throughout optimization are reported in table 1. It is interesting to notice how fixing covariance parameters $\boldsymbol{\theta}$ leads to comparable performance to the case where they are learned.

Table 1: Results on the proposed CNN+GP(SORF) when fixing or learning covariance parameters $\boldsymbol{\theta}$. All results were obtained on MNIST, CIFAR10, and CIFAR100 without subsampling the data. Please refer to table 1 in the main paper for details on the convolutional structure corresponding to SHALLOW and DEEP.

	SHALLOW				DEEP			
	MNIST		CIFAR10		CIFAR10		CIFAR100	
Metrics	Fixed	Learned	Fixed	Learned	Fixed	Learned	Fixed	Learned
ERR	0.006	0.005	0.203	0.192	0.113	0.115	0.352	0.359
MNLL	0.018	0.018	0.610	0.584	0.348	0.355	1.264	1.287
ECE	0.002	0.003	0.015	0.010	0.051	0.054	0.050	0.054
BRIER	0.009	0.008	0.288	0.271	0.170	0.173	0.466	0.478

3 Variational inference of filters in GPDNN

In this section we report results when applying variational inference on the weights in GPDNN (Bradshaw et al., 2017). In order to do this, we implemented MCD for the convolutional parameters, similarly to what

presented in the main paper for our CNN+GP(RF) model.

Table 2: Results on the proposed CNN+GP(SORF) vs GPDNN when inferring convolutional parameters using MCD. All results were obtained on MNIST, CIFAR10, and CIFAR100 without subsampling the data. Please refer to table 1 in the main paper for details on the convolutional structure corresponding to SHALLOW and DEEP.

	SHALLOW				DEEP			
	MNIST		CIFAR10		CIFAR10		CIFAR100	
Metrics	CNN+GP(RF)	GPDNN	CNN+GP(RF)	GPDNN	CNN+GP(RF)	GPDNN	CNN+GP(RF)	GPDNN
ERR	0.005	0.005	0.172	0.172	0.111	0.190	0.351	0.820
MNLL	0.014	0.019	0.535	0.531	0.344	0.675	1.255	8.606
ECE	0.004	0.005	0.012	0.012	0.051	0.036	0.050	0.527
BRIER	0.0071	0.008	0.245	0.244	0.168	0.278	0.466	1.268

The results in table 2 indicate that this improves the calibration and accuracy of GPDNN compared to optimizing the filters. In the case of a shallow convolutional architecture, the performance of CNN+GP(RF) and GPDNN are comparable, although in the deeper case CNN+GP(RF) achieves better performance. This supports the intuition that inferring convolutional parameters, ranther than optimizing them, leads to considerable improvements in calibration.

4 Reliability diagrams

In this section, we report the reliability diagram and histogram of predictive output for all methods with various datasets, i.e CIFAR10 and CIFAR100 and convolutional architectures, i.e LeNet and RESNET. We use the best configuration for CGP according to the implementation released by the Authors. In each figure, rows correspond with the dataset and convolutional architecture, while the column refer to the training size. After the training phase, all models are evaluated on the entire testing set. The number of bins used to draw the reliability diagram is 20.

In each subfigure, the dashed line indicates perfect calibration. The horizontal axis is the softmax output ranging from 0 to 1. The vertical axis indicates accuracy rate for the red line or frequency for the green bars. The red dot is the real average accuracy at each bin, while the line segments at the red dots refer to the standard deviation of the accuracies. The green bar is the average frequency histogram at each bin of softmax values. The experiments of GPDNN, CGP, MCD-CIFAR10-LeNet and CNN+GP(RF) are repeated three times.

Having observed these figures, we see that regularizing convolutional filters has a huge impact on calibration. From figures 2, 3, 4 and 5 we see that CNNs and the previous combinations of GPs and CNNs are miscalibrated. From figure 7 and 8, instead, we see that Bayesian CNNs improve the reliability of the prediction, which is comparable with post-calibration.

It seems that there is a correlation between the histogram of predictive output and the reliability line. When the histogram is skewed to the right, the corresponding classifier is poorly calibrated.

Post calibration, MCD and CNN+GP(RF) (our method) are able to yeild calibrated classification.

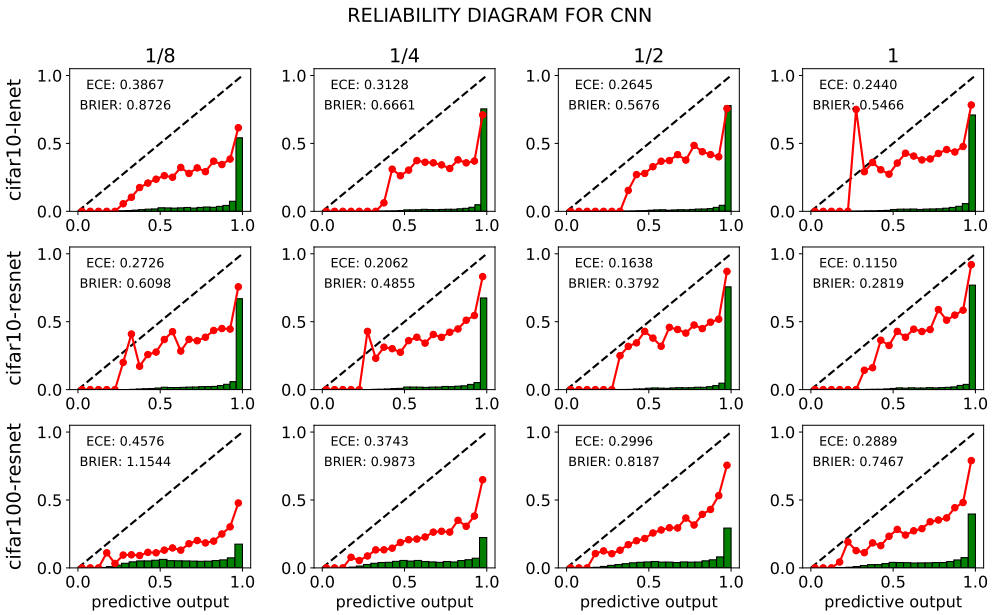

Figure 2: Reliability diagrams for CNN

5 Visualization of Convolutional Filters

We investigate the impact of our approach on convolutional filters. In figure 9, we show the filters that plain CNNs, Bayesian CNNs (with MCD) and our CNN+GP(RF) learn in the first and second convolutional layer when applied to the CIFAR10 data set. The CNN in the experiment includes two convolutional layers with depth of 16 and receptive field of 11×11. The convolutional features are 4096-dimensional vector and these features are fed into a fully connected layer with 256 hidden units. The number of random features in CNN+GP(RF) is 256. Each row in the figure corresponds to an input channel, while each column is for an output channel. The figure shows that CNN+GP(RF) learns filters that are similar to MCD, but quite different to plain CNNs.

References

J. Bradshaw, Alexander, and Z. Ghahramani. Adversarial Examples, Uncertainty, and Transfer Testing Robustness in Gaussian Process Hybrid Deep Networks, July 2017. arXiv:1707.02476.

Y. Gal and Z. Ghahramani. Dropout As a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1050–1059. JMLR.org, 2016a.

Y. Gal and Z. Ghahramani. Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference, Jan. 2016b. arXiv:1506.02158.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6402–6413. Curran Associates, Inc., 2017.

M. Lázaro-Gredilla, J. Quinonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal. Sparse Spectrum Gaussian Process Regression. *Journal of Machine Learning Research*, 11:1865–1881, 2010.

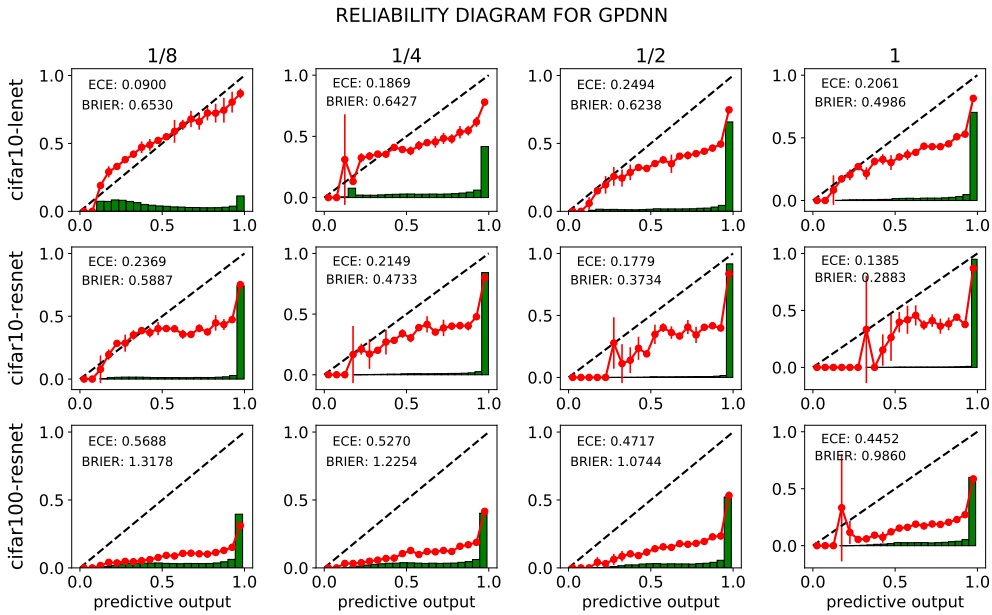

Figure 3: Reliability diagrams for GPDNN

A. Rahimi and B. Recht. Random Features for Large-Scale Kernel Machines. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1177–1184. Curran Associates, Inc., 2008.

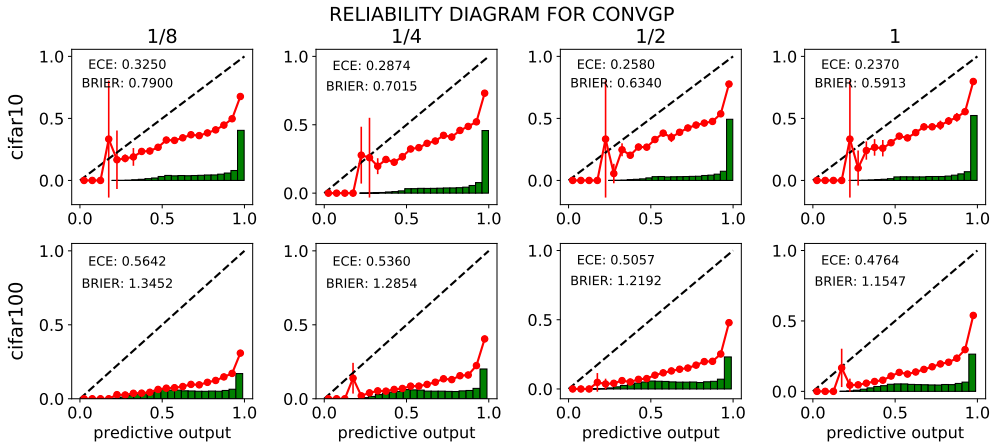

Figure 4: Reliability diagrams for CGP

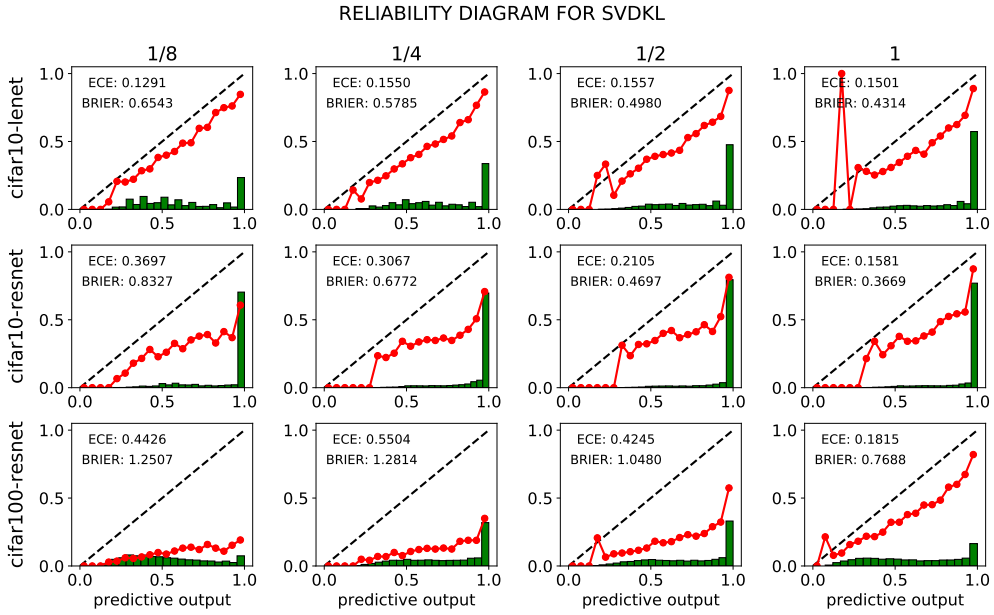

Figure 5: Reliability diagrams for SVDKL

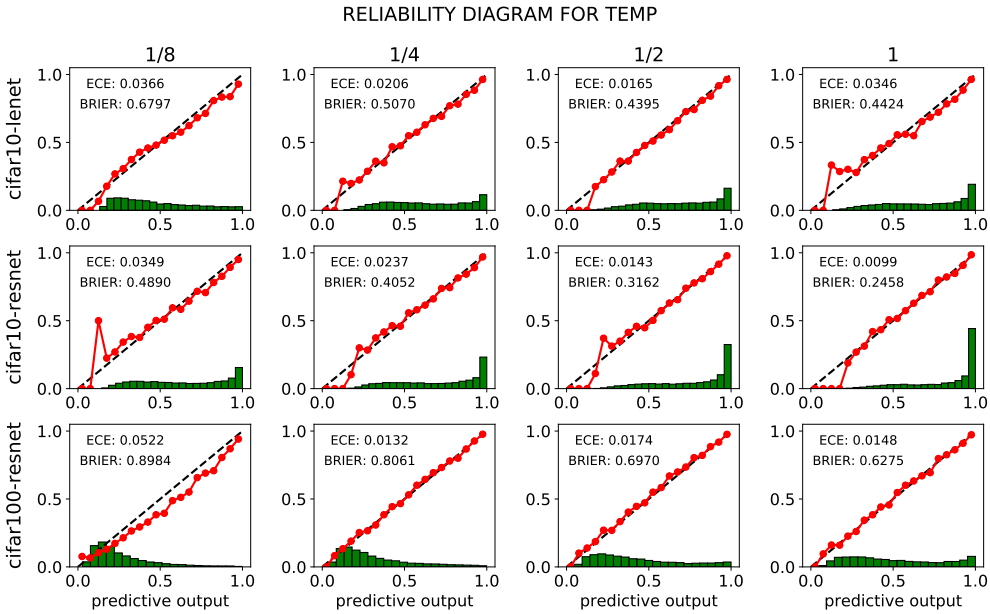

Figure 6: Reliability diagrams for CNN+CAL

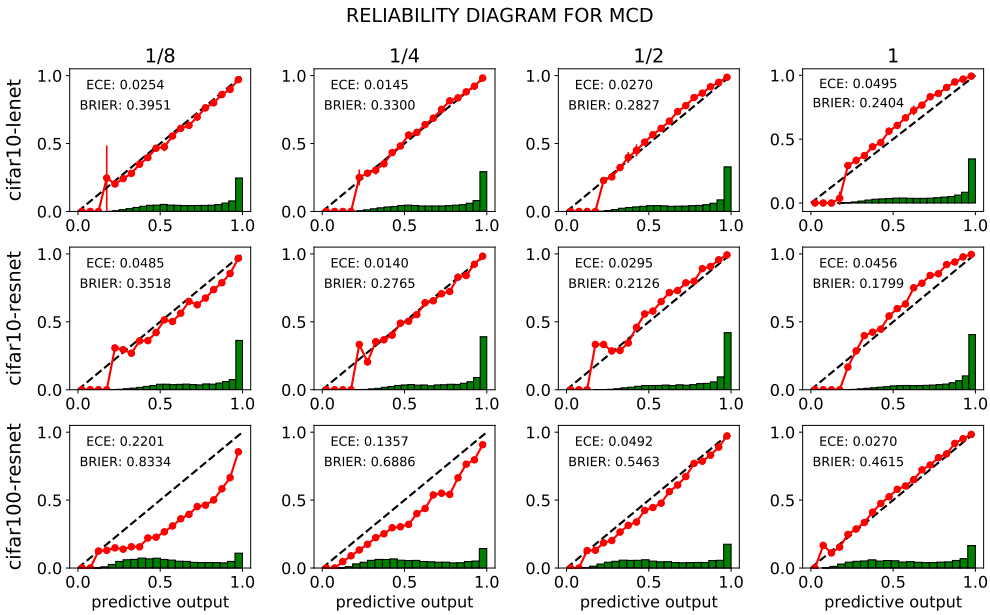

Figure 7: Reliability diagrams for MCD

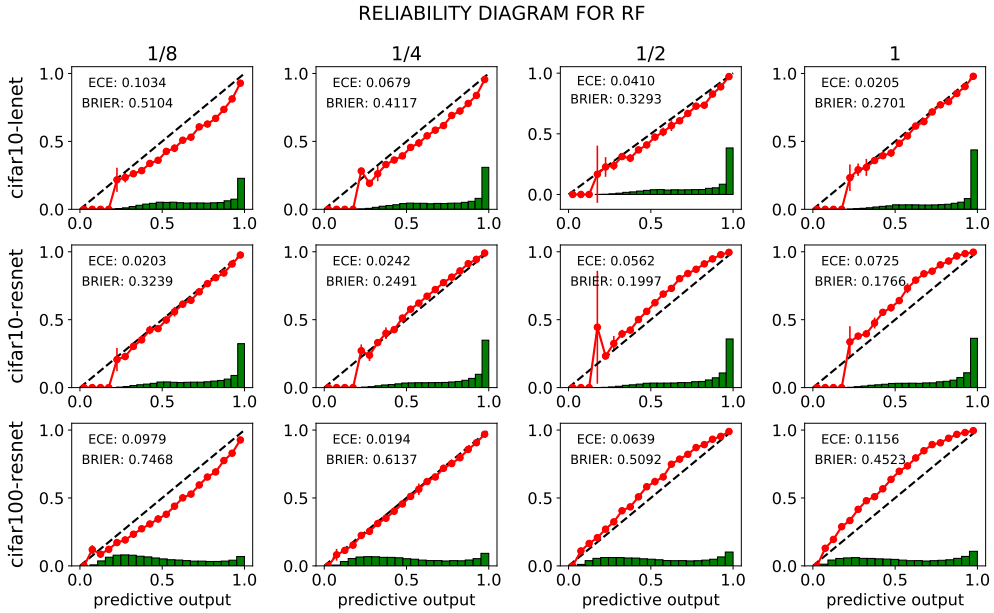

Figure 8: Reliability diagrams for CNN+GP(RF)

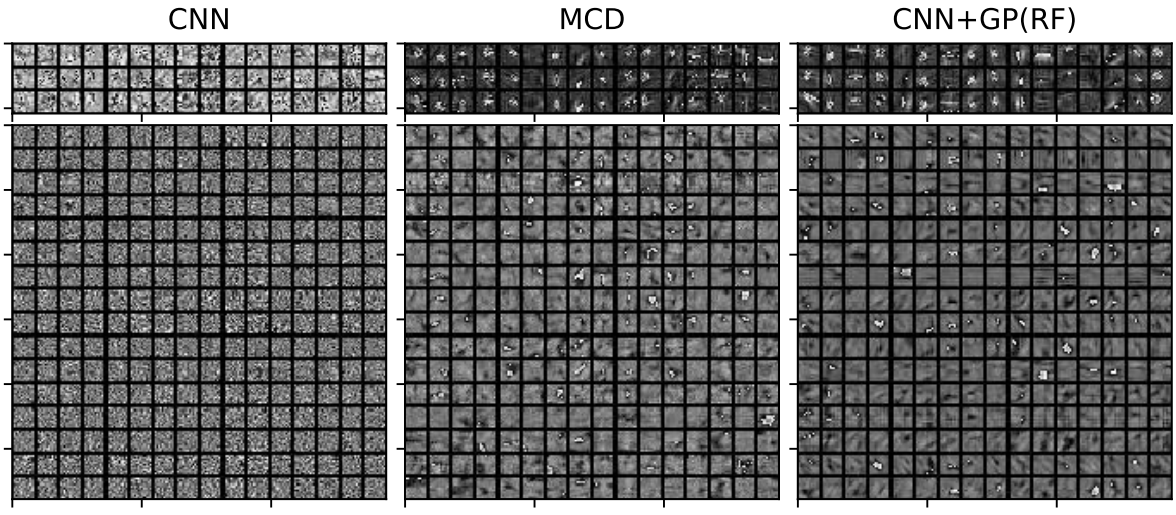

Figure 9: Visualization of the first and second convolutional filters of CNN, MCD and CNN+GP(RF)- CIFAR10 data set