Supplementary Material: Non-Parametric Calibration for Classification

This is the supplementary material for the paper: "Non-Parametric Calibration for Classification", by Jonathan Wenger, Hedvig Kjellström and Rudolph Triebel.

S1 OVER- AND UNDERCONFIDENCE

The notions over- and underconfidence

$$o(f) = \mathbb{E}\left[\hat{\mathrm{z}} \mid \hat{\mathrm{y}} \neq \mathrm{y}\right] \quad u(f) = \mathbb{E}\left[1 - \hat{\mathrm{z}} \mid \hat{\mathrm{y}} = \mathrm{y}\right],$$

as introduced in (2), quantify the amount of information contained in the uncertainty estimate of a classifier f about the true class. Note, that their definitions are decoupled from the accuracy of f.

S1.1 Definition Subtleties

Differing from the corresponding intuitive notions, a classifier can simultaneously be over- and underconfident to varying degree. In particular $o(f) > 0$ does not imply $u(f) = 0$, and neither does the reverse. Consider the following, where the true posterior $p(\mathrm{y} \mid \mathbf{x})$ assigns 75% confidence to one class and the remaining 25% uniformly across all other classes. Assume now the classifier assigns 90% confidence to the given class and 10% uniformly to the remaining classes on the entire input space. Intuition would dictate this classifier to not necessarily be underconfident, but by definition $u(f) = 0.1$ and $o(f) = 0.9$. This is because over- and underconfidence are not a statistical distance between posterior distributions, but describe the difference between the classifier's posterior distribution and an unobserved deterministic underlying relationship $(\mathbf{x}, \mathrm{y})$. Hence, aleatoric and epistemic uncertainty contained in the data distribution influence the over- and underconfidence of a classifier. This can be seen by computing the over- and underconfidence for the true posterior distribution in the above example giving $u(f) = 0.25$ and $o(f) = 0.75$. Obtaining lower over- or underconfidence in this case is only possible by sacrificing one or the other.

S1.2 Proof of Theorem 1

We give a proof for the calibration error bound to the weighted absolute difference between over- and underconfidence as stated in Theorem 1 below.

Proof. By linearity of expectation and the law of total expectation it holds that

$$\begin{aligned}\mathbb{E}\left[\hat{\mathrm{z}}\right] &= \mathbb{E}\left[\hat{\mathrm{z}} + \mathbb{E}\left[1_{\hat{\mathrm{y}}=\mathrm{y}} \mid \hat{\mathrm{z}}\right] - \mathbb{E}\left[1_{\hat{\mathrm{y}}=\mathrm{y}} \mid \hat{\mathrm{z}}\right]\right] \\ &= \mathbb{E}\left[\hat{\mathrm{z}} - \mathbb{E}\left[1_{\hat{\mathrm{y}}=\mathrm{y}} \mid \hat{\mathrm{z}}\right]\right] + \mathbb{P}(\hat{\mathrm{y}} = \mathrm{y}).\end{aligned}$$

Conversely, by decomposing the average confidence we have

$$\begin{aligned}\mathbb{E}\left[\hat{\mathrm{z}}\right] &= \mathbb{E}\left[\hat{\mathrm{z}} \mid \hat{\mathrm{y}} \neq \mathrm{y}\right]\mathbb{P}(\hat{\mathrm{y}} \neq \mathrm{y}) + \mathbb{E}\left[\hat{\mathrm{z}} \mid \hat{\mathrm{y}} = \mathrm{y}\right]\mathbb{P}(\hat{\mathrm{y}} = \mathrm{y}) \\ &= \mathbb{E}\left[\hat{\mathrm{z}} \mid \hat{\mathrm{y}} \neq \mathrm{y}\right]\mathbb{P}(\hat{\mathrm{y}} \neq \mathrm{y}) + \\ &\quad (1 - \mathbb{E}\left[1 - \hat{\mathrm{z}} \mid \hat{\mathrm{y}} = \mathrm{y}\right])\mathbb{P}(\hat{\mathrm{y}} = \mathrm{y}) \\ &= o(f)\mathbb{P}(\hat{\mathrm{y}} \neq \mathrm{y}) + (1 - u(f))\mathbb{P}(\hat{\mathrm{y}} = \mathrm{y}).\end{aligned}$$

Combining the above we obtain

$$\mathbb{E}\left[\hat{\mathrm{z}} - \mathbb{E}\left[1_{\hat{\mathrm{y}}=\mathrm{y}} \mid \hat{\mathrm{z}}\right]\right] = o(f)\mathbb{P}(\hat{\mathrm{y}} \neq \mathrm{y}) - u(f)\mathbb{P}(\hat{\mathrm{y}} = \mathrm{y}).$$

Now, since $h(x) = |x|^p$ is convex for $1 \leq p < \infty$, we have by Jensen's inequality

$$\left|\mathbb{E}\left[\hat{\mathrm{z}} - \mathbb{E}\left[1_{\hat{\mathrm{y}}=\mathrm{y}} \mid \hat{\mathrm{z}}\right]\right]\right|^p \leq \mathbb{E}\left[\left|\hat{\mathrm{z}} - \mathbb{E}\left[1_{\hat{\mathrm{y}}=\mathrm{y}} \mid \hat{\mathrm{z}}\right]\right|^p\right]$$

and finally by Hölder's inequality with $1 \leq p < q \leq \infty$ it follows that

$$\begin{aligned}\mathrm{ECE}_p &= \mathbb{E}\left[\left|\hat{\mathrm{z}} - \mathbb{E}\left[1_{\hat{\mathrm{y}}=\mathrm{y}} \mid \hat{\mathrm{z}}\right]\right|^p\right]^{\frac{1}{p}} \\ &\leq \mathbb{E}\left[\left|\hat{\mathrm{z}} - \mathbb{E}\left[1_{\hat{\mathrm{y}}=\mathrm{y}} \mid \hat{\mathrm{z}}\right]\right|^q\right]^{\frac{1}{q}} = \mathrm{ECE}_q,\end{aligned}$$

which concludes the proof. □

S2 DETAILED INFERENCE AND CALIBRATION

We give a more detailed exposition of GP calibration inference. We begin by describing the derivation of the bound on the marginal log-likelihood in (6).

S2.1 Bound on the Marginal Log-Likelihood

This subsection follows Hensman et al. (2015) and is adapted for our specific inverse link function and likelihood. Consider the following bound, derived by marginalization and Jensen's inequality.

$$\begin{aligned}\ln p(\mathbf{y} \mid \mathbf{u}) &= \ln \mathbb{E}_{p(\mathbf{g}\mid\mathbf{u})}\left[p(\mathbf{y} \mid \mathbf{g})\right] \\ &\geq \mathbb{E}_{p(\mathbf{g}\mid\mathbf{u})}\left[\ln p(\mathbf{y} \mid \mathbf{g})\right]\end{aligned} \tag{7}$$

We then substitute (7) into the lower bound to the evidence (ELBO) as follows

$$\begin{aligned} \ln p(\mathbf{y}) &= \mathrm{KL}\left[q(\mathbf{u}) \| p(\mathbf{u} \mid \mathbf{y})\right] + \mathrm{ELBO}(q(\mathbf{u})) \\ &\geq \mathrm{ELBO}(q(\mathbf{u})) \\ &= \mathbb{E}_{q(\mathbf{u})}\left[\ln p(\mathbf{y}, \mathbf{u})\right] - \mathbb{E}_{q(\mathbf{u})}\left[\ln q(\mathbf{u})\right] \\ &= \mathbb{E}_{q(\mathbf{u})}\left[\ln p(\mathbf{y} \mid \mathbf{u})\right] - \mathrm{KL}\left[q(\mathbf{u}) \| p(\mathbf{u})\right] \\ &\geq \mathbb{E}_{q(\mathbf{u})}\left[\mathbb{E}_{p(\mathbf{g}\mid\mathbf{u})}\left[\ln p(\mathbf{y} \mid \mathbf{g})\right]\right] \\ &\qquad\qquad - \mathrm{KL}\left[q(\mathbf{u}) \| p(\mathbf{u})\right] \\ &= \mathbb{E}_{q(\mathbf{g})}\left[\ln p(\mathbf{y} \mid \mathbf{g})\right] - \mathrm{KL}\left[q(\mathbf{u}) \| p(\mathbf{u})\right] \\ &= \sum_{n=1}^{N} \mathbb{E}_{q(\mathbf{g}_n)}\left[\ln p(\mathrm{y}_n \mid \mathbf{g}_n)\right] \\ &\qquad\qquad - \mathrm{KL}\left[q(\mathbf{u}) \| p(\mathbf{u})\right], \end{aligned} \tag{8}$$

where $q(\mathbf{g}) \coloneqq \int p(\mathbf{g} \mid \mathbf{u}) q(\mathbf{u})\, d\mathbf{u}$ and the last equality holds by independence of the calibration data. By (5) and the properties of Gaussians we obtain

$$p(\mathbf{g} \mid \mathbf{u}) = \mathcal{N}(\mathbf{g} \mid \boldsymbol{\mu}_{\mathbf{g}\mid\mathbf{u}}, \boldsymbol{\Sigma}_{\mathbf{g}\mid\mathbf{u}})$$

such that

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{g}\mid\mathbf{u}} &= \boldsymbol{\mu}_{\mathbf{g}} + \boldsymbol{\Sigma}_{\mathbf{g},\mathbf{u}} \boldsymbol{\Sigma}_{\mathbf{u}}^{-1} (\mathbf{u} - \boldsymbol{\mu}_{\mathbf{u}}) \\ \boldsymbol{\Sigma}_{\mathbf{g}\mid\mathbf{u}} &= \boldsymbol{\Sigma}_{\mathbf{g}} - \boldsymbol{\Sigma}_{\mathbf{g},\mathbf{u}} \boldsymbol{\Sigma}_{\mathbf{u}}^{-1} \boldsymbol{\Sigma}_{\mathbf{g},\mathbf{u}}^{\top}. \end{aligned}$$

Let $q(\mathbf{u}) = \mathcal{N}(\mathbf{u} \mid \boldsymbol{m}, \boldsymbol{S})$ and $\boldsymbol{A} \coloneqq \boldsymbol{\Sigma}_{\mathbf{g},\mathbf{u}} \boldsymbol{\Sigma}_{\mathbf{u}}^{-1}$, then

$$\begin{aligned} q(\mathbf{g}) &\coloneqq \int \underbrace{p(\mathbf{g} \mid \mathbf{u}) q(\mathbf{u})}_{q(\mathbf{g},\mathbf{u})}\, d\mathbf{u} \\ &= \mathcal{N}(\mathbf{g} \mid \boldsymbol{\mu}_{\mathbf{g}} + \boldsymbol{A}(\boldsymbol{m} - \boldsymbol{\mu}_{u}),\, \boldsymbol{\Sigma}_{\mathbf{g}} + \boldsymbol{A}(\boldsymbol{S} - \boldsymbol{\Sigma}_{\mathbf{u}})\boldsymbol{A}^{\top}). \end{aligned}$$

as $q(\mathbf{g}, \mathbf{u})$ is normally distributed. To compute the expectations in (8) we only need to consider the K-dimensional marginals

$$q(\mathbf{g}_n) = \int p(\mathbf{g}_n \mid \mathbf{u}) q(\mathbf{u})\, d\mathbf{u} = \mathcal{N}(\mathbf{g}_n \mid \boldsymbol{\varphi}_n, \boldsymbol{C}_n).$$

S2.2 Approximation of the Expectation Terms

In order to obtain the variational objective (8) we need to compute the expected value terms for our intractable likelihood (4). To do so, we use a second order Taylor approximation of

$$h(\mathbf{g}_n) \coloneqq \ln p(\mathrm{y}_n \mid \mathbf{g}_n) = \ln \frac{\exp\left(\mathrm{g}_{n\mathrm{y}_n}\right)}{\sum_{k=1}^{K} \exp(\mathrm{g}_{nk})}$$

at $\mathbf{g}_n = \boldsymbol{\varphi}_n$. The Hessian of the log-softargmax is given by

$$\begin{aligned} D_{\mathbf{g}_n}^2 h(\mathbf{g}_n) &= D_{\mathbf{g}_n}^2 \ln \sigma(\mathbf{g}_n)_{\mathrm{y}_n} \\ &= \sigma(\mathbf{g}_n)\sigma(\mathbf{g}_n)^{\top} - \mathrm{diag}(\sigma(\mathbf{g}_n)). \end{aligned}$$

Note this expression does not depend on y_n. We obtain by using $\boldsymbol{x}^{\top}\boldsymbol{M}\boldsymbol{x} = \mathrm{tr}\big(\boldsymbol{x}^{\top}\boldsymbol{M}\boldsymbol{x}\big)$, the linearity of the trace and its invariance under cyclic permutations, that

$$\begin{aligned} &\mathbb{E}_{q(\mathbf{g}_n)}\left[\ln p(\mathrm{y}_n \mid \mathbf{g}_n)\right] = \mathbb{E}_{q(\mathbf{g}_n)}\left[h(\mathbf{g}_n)\right] \\ &\quad \approx \mathbb{E}_{q(\mathbf{g}_n)}\Big[h(\boldsymbol{\varphi}_n) + D_{\mathbf{g}_n} h(\boldsymbol{\varphi}_n)^{\top}(\mathbf{g}_n - \boldsymbol{\varphi}_n) \\ &\qquad\qquad + \frac{1}{2}(\mathbf{g}_n - \boldsymbol{\varphi}_n)^{\top} D_{\mathbf{g}_n}^2 h(\boldsymbol{\varphi}_n)(\mathbf{g}_n - \boldsymbol{\varphi}_n)\Big] \\ &\quad = h(\boldsymbol{\varphi}_n) + \frac{1}{2}\mathbb{E}_{q(\mathbf{g}_n)}\Big[(\mathbf{g}_n - \boldsymbol{\varphi}_n)^{\top}\big(\sigma(\boldsymbol{\varphi}_n)\sigma(\boldsymbol{\varphi}_n)^{\top} \\ &\qquad\qquad - \mathrm{diag}(\sigma(\boldsymbol{\varphi}_n))\big)(\mathbf{g}_n - \boldsymbol{\varphi}_n)\Big] \\ &\quad = h(\boldsymbol{\varphi}_n) + \frac{1}{2}\,\mathrm{tr}\Big[\mathbb{E}_{q(\mathbf{g}_n)}\big[(\mathbf{g}_n - \boldsymbol{\varphi}_n)(\mathbf{g}_n - \boldsymbol{\varphi}_n)^{\top}\big] \\ &\qquad\qquad \big(\sigma(\boldsymbol{\varphi}_n)\sigma(\boldsymbol{\varphi}_n)^{\top} - \mathrm{diag}(\sigma(\boldsymbol{\varphi}_n))\big)\Big] \\ &\quad = h(\boldsymbol{\varphi}_n) + \frac{1}{2}\,\mathrm{tr}\big[\boldsymbol{C}_n\left(\sigma(\boldsymbol{\varphi}_n)\sigma(\boldsymbol{\varphi}_n)^{\top} - \mathrm{diag}(\sigma(\boldsymbol{\varphi}_n))\right)\big] \\ &\quad = h(\boldsymbol{\varphi}_n) + \frac{1}{2}\big(\mathrm{tr}\big[\sigma(\boldsymbol{\varphi}_n)^{\top}\boldsymbol{C}_n\sigma(\boldsymbol{\varphi}_n)\big] \\ &\qquad\qquad - \mathrm{tr}[\boldsymbol{C}_n \mathrm{diag}(\sigma(\boldsymbol{\varphi}_n))]\big) \\ &\quad = h(\boldsymbol{\varphi}_n) + \frac{1}{2}\big(\sigma(\boldsymbol{\varphi}_n)^{\top}\boldsymbol{C}_n\sigma(\boldsymbol{\varphi}_n) - \mathrm{diag}(\boldsymbol{C}_n)^{\top}\sigma(\boldsymbol{\varphi}_n)\big), \end{aligned}$$

which can be computed in $\mathcal{O}(K^2)$. This is apparent when expressing the term inside the parentheses as a double sum over K terms.

S3 EXPERIMENT DETAILS

In this section, we elaborate on the experiments in Section 4. We discuss the approximation to the expected calibration error, hyperparameter choices and give runtime measurements and classification accuracy of the performed calibration experiments. Finally, we show some additional visualizations of latent functions and reliability diagrams.

S3.1 Choice of Number of Bins

In practice, we estimate the calibration error as suggested by Naeini et al. (2015) by introducing a fixed uniform binning $0 = \vartheta_0 < \vartheta_1 < \cdots < \vartheta_B = 1$ such that

$$\mathrm{ECE}_p \approx \frac{1}{B}\left(\sum_{b=1}^{B} \left|\bar{\hat{\mathrm{z}}}_b - \mathrm{acc}_b\right|^p\right)^{\frac{1}{p}}, \tag{9}$$

where

$$\bar{\hat{\mathrm{z}}}_b = \frac{1}{N_b} \sum_{\vartheta_{b-1} < \hat{\mathrm{z}} \leq \vartheta_b} \hat{\mathrm{z}}$$

is the mean confidence in bin b,

$$\mathrm{acc}_b = \frac{1}{N_b} \sum_{\vartheta_{b-1} < \hat{\mathrm{z}} \leq \vartheta_b} 1_{\hat{\mathrm{y}} = \mathrm{y}}$$

the accuracy in bin b and N_b the number of samples in bin b, such that $N = \sum_{b=1}^{B} N_b$. In previous work introducing the calibration error (Naeini et al., 2015) and in (Guo et al., 2017) the discretization of the expected calibration error ECE_1 uses 15 equally spaced bins.

As described in the supplementary material in (Guo et al., 2017), the empirical estimate is a good approximation to the expected calibration error for N and B sufficiently large. However, in the infinite sample case the larger the bin size B the tighter the empirical estimate lower bounds the ECE_1 (Kumar et al., 2019). This is due to the fact that over- and underestimation of uncertainty within one bin cancel each other out. Hence, *using too few bins can underestimate the expected calibration error.* Kumar et al. (2019) also observe this in practice. This phenomenon is particularly prevalent for CNNs as most of their confidence predictions fall into one or two bins for $B = 15$. This is also the case in our experiments and can be seen in the histograms in Figures S3 to S5. We observed the aforementioned underestimation of calibration error with 15 bins for some data set and model combinations in our experiments. To mitigate this problem we deliberately chose $B = 100$ in this work. The number of bins is limited by the number of test samples we have available, as within-bin-variance increases with the number of bins. Further work needs to be done to determine the properties of the estimator to ECE_p and the ideal discretization for a given number of samples.

S3.2 Implementation and Hyperparameters

All experiments were performed using the `pycalib` package available at

`https://github.com/JonathanWenger/pycalib`.

When calibrating models returning logits, we used a sum kernel for the one-dimensional latent Gaussian process given by

$$\begin{aligned} k(x, x') &= k_{\text{RBF}}(x, x') + k_{\text{noise}}(x, x') \\ &= \sigma^2 \exp\left(-\frac{\|x - x'\|^2}{2l^2}\right) + \sigma_{\text{noise}}^2 \delta_{x,x'} \end{aligned}$$

where k_{RBF} is an exponentiated quadratic and k_{noise} a white noise kernel. The kernel parameters were initialised as $\sigma = 1, l = 10$ and $\sigma_{\text{noise}} = 0.01$. For GP inference we used $M = 10$ inducing points. The `gpflow` package with the `scipy` implementation of the L-BFGS optimizer was used to find inducing points and kernel hyperparameters in the variational inference procedure. For calibration we used $Q = 100$ Monte-Carlo samples to compute the posterior distribution.

S3.3 Binary Experiments

For the binary calibration experiments we used the following data sets with indicated train, calibration and test splits:

- KITTI (Geiger et al., 2012, Narr et al., 2016): Stream-based urban traffic scenes with features (Himmelsbach et al., 2009) from segmented 3D point clouds. 8 or 2 classes, train: 16000, calibration: 1000, test: 8000.
- PCam (Veeling et al., 2018): Histopathologic scans of (metastatic) tissue from lymph node sections converted to grayscale. 2 classes, train: 22768, calibration: 1000, test: 9000.

The resulting average calibration error across random samples of calibration data sets is shown in Table S3. Isotonic regression performed the best in terms of calibration on KITTI. However most methods performed well and often within one standard deviation of each other for both binary data sets. Hence, for binary problems a simple binary calibration method may suffice.

S3.4 Multi-class Experiments

We provide more detailed calibration results of our multi-class experiment explained in Section 4.1 in Table S5. We show the average calibration error (ECE_1), including standard deviation, of the presented calibration methods and the GPcalib mean approximation on all data set and model combinations.

S3.5 Accuracy

The accuracy from the binary and multi-class experiments described in Section 4 is given in Table S2 and Table S4, respectively. For the binary experiments accuracy is mostly unaffected across classifiers and even improves in some instances. Only Bayesian binning into quantiles suffers from a noticable drop in accuracy for random forests. Somewhat surprisingly, for the simple neural network all binary methods actually improve upon accuracy.

In the multi-class case we see that accuracy is severely affected for binary methods extended in a one-vs-all fashion for the ImageNet data set, disqualifying them from use. Both temperature scaling and GP calibration preserve accuracy across models and data sets.

S3.6 Wall-Clock Runtime

We provide wall-clock runtime for each data set considered in our experiments from Section 4. As the runtime was very similar across classifiers we show average runtime per data set across models in Table S1. Note that wall-clock runtime is highly dependent on the specific machine used for computation. In our case we used a 12-core desktop computer with a GeForce RTX 2080 Ti graphics card for all experiments in this work. Time for inference and calibration scales close to linear with the number of classes for almost all calibration methods. There seems to be some unavoidable overhead for calibration irrespective of classes as can be seen when looking at binary problems. GPcalib takes more time for parameter inference than other methods. This is due to the fact that we need to perform approximate inference and since GPcalib is the only method taking calibration uncertainty into account. Potentially speed-up via the use of GPUs for accelerated matrix computations for our `gpflow`-based implementation of GPcalib is possible. However, in practice the times taken for inference and calibration are significantly less than those for training and prediction of the underlying classifier. Hence, *a classifier can be calibrated with little added time cost.*

S3.7 Additional Latent Function Plots

Further examples of latent calibration maps from GPcalib and temperature scaling are given in this section. Figure S1 shows latent functions from a single CV run of the MNIST experiments in Section 4.1.When applying GPcalib to classifiers outputting probability scores, the ln prior corresponds to the assumption of the underlying model being calibrated. Note, that temperature scaling was not designed to be used on probability scores, but on logits, this causes its latent function to have very small slope. Both in the case of probability scores and logits, by definition of GPcalib and temperature scaling any constant shift of the latent function results in the same uncertainty estimates. The shown plots should be interpreted with this in mind. For models which are very miscalibrated, the latent function of GPcalib demonstrates a high degree of non-linearity and deviation from the ln prior, e.g. in the case of AdaBoost. When the underlying model is already close to being calibrated as in the case of the 1-layer neural network, the resulting latent GP does not deviate very much from the prior mean.

In the multi-class calibration experiment on ImageNet, we observed better calibration of GPcalib for higher accuracy CNNs. In Figure S2, we show additional latent functions obtained from the experiment. For VGG19 and DenseNet-201, the underlying network had comparably low calibration error to begin with. Both temperature scaling and GPcalib did not improve calibration significantly. This is reflected in latent space, where they do not deviate much from the identity map. However, in the case of SE-ResNeXt-50, SE-ResNeXt-101, SENet-154 and NASNet-A-Large GPcalib improved calibration noticably over the baseline and temperature scaling. Again, this improvement corresponds to a large change from the identity map in its latent function.

S3.8 Reliability Diagrams

Reliability diagrams (DeGroot and Fienberg, 1983, Niculescu-Mizil and Caruana, 2005a) visualize the degree of calibration of a model. They consist of a plot comparing confidence estimates with accuracy for a given binning of $[0, 1]$ and a histogram of confidence estimates. They relate to the ECE_1 in the following way. The gray deviation from the diagonal in Figures S3 to S5 weighted by the histogram below equals the estimate of the ECE_1 for a given binning. Here, for visualization purposes we chose 15 bins instead of 100 as in our experiments. For such a binning, most confidence estimates fall into one or two bins, leading to the estimation problem described in Section S3.1. When interpreting reliability diagrams keep in mind that bins with a low number of samples have high variance in their per-bin-accuracy. We show some reliability diagrams for CNNs on ImageNet from our experiments in Section 4.1. As is the case for most high-accuracy CNNs, Resnet-152 and PolyNet shown in Figures S3 and S4 mostly predict with very high confidence. Further, they generally predict higher confidence than accuracy, which is reflected in their high overconfidence. This observation is consistent with previous work on overconfidence of neural networks (Lakshminarayanan et al., 2017, Guo et al., 2017, Hein et al., 2019). Interestingly, PNASNet-5-Large (see Figure S5) is actually underconfident. This also holds true for SENet-154 and NASNet-A-Large, which demonstrated the highest accuracy on ImageNet in our experiments.

Table S1: *Average wall-clock runtime of experiments.* We show average time in seconds of inference on the calibration sets and calibration on the test sets. Times are averaged for each data set across classifiers and 10 Monte-Carlo cross validation folds, since variance between classifiers on the same data set was small.

		one-vs-all						
Mode	Data Set	Platt	Isotonic	Beta	BBQ	Temp.	GPcalib	GPcalib mean appr.
Inference	KITTI	0.0017	0.0004	0.0032	0.0994	0.0053	1.0517	1.0441
	PCam	0.0020	0.0007	0.0026	0.0923	0.0063	1.0653	1.0492
	MNIST	0.0245	0.0055	0.0478	0.8589	0.0129	2.2112	2.1596
	CIFAR-100	0.3181	0.0602	0.2052	9.0458	0.0277	20.6564	20.4714
	ImageNet	3.0911	0.9436	2.2291	61.3060	0.3395	259.7305	258.0205
Calibration	KITTI	0.0002	0.0003	0.0022	0.1334	0.0006	0.1625	0.1227
	PCam	0.0003	0.0004	0.0021	0.1502	0.0008	0.1610	0.1171
	MNIST	0.0043	0.0046	0.1126	1.5180	0.0026	0.2610	0.1483
	CIFAR-100	0.1276	0.1269	0.3202	14.8716	0.0210	2.4482	0.3367
	ImageNet	8.3651	8.3005	12.7822	113.6108	0.2369	51.3806	3.1317

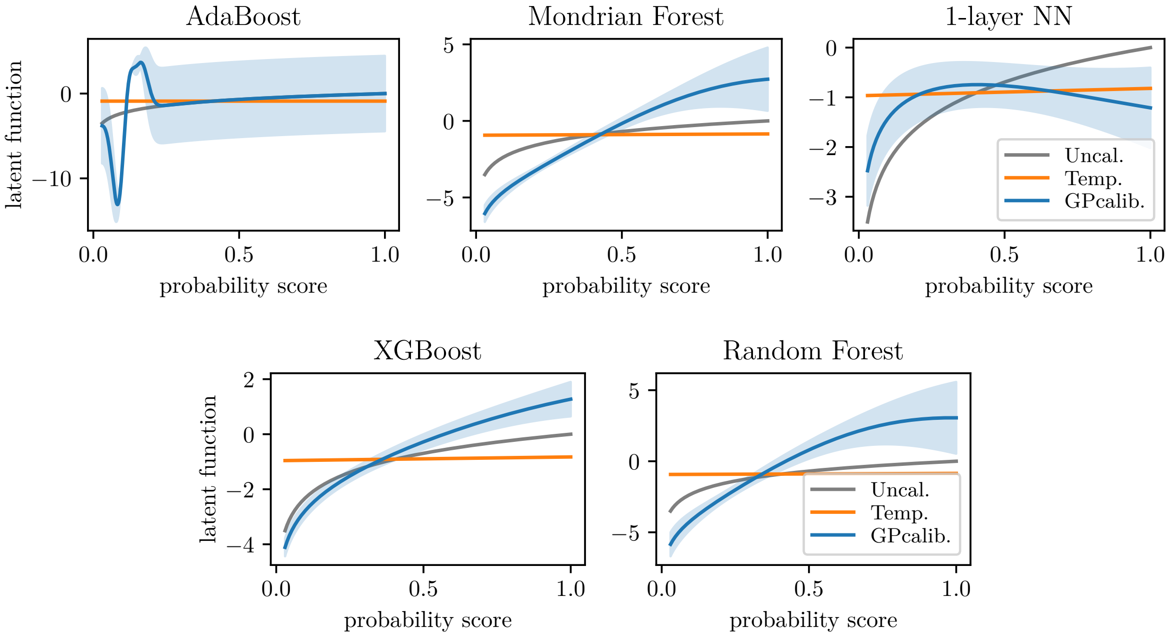

Figure S1: *Illustration of latent functions on probability scores.* Latent functions of GPcalib and temperature scaling for probability scores from classifiers on MNIST. GPcalib uses the ln prior corresponding to the prior assumption of the classifier being calibrated. For AdaBoost we can see in Table 1, that remedying the large calibration error is only handled well by GPcalib. This corresponds to a large deviation from the prior in latent space.

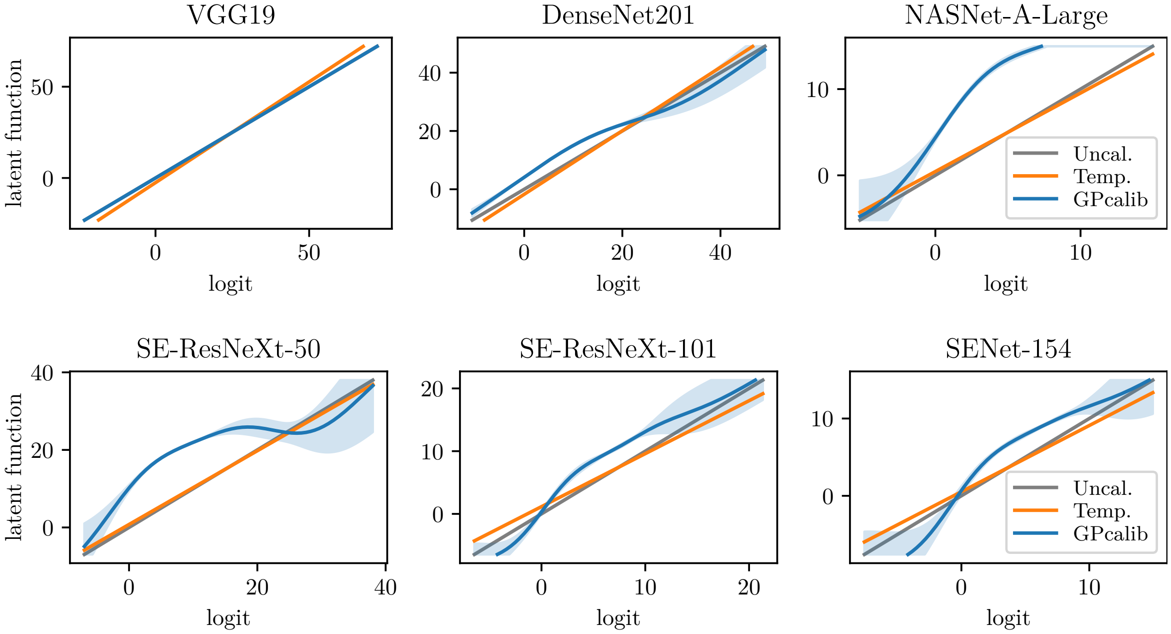

Figure S2: *Illustration of non-linear latent functions in logit-space.* Additional latent maps of temperature scaling and GPcalib from our experiments on ImageNet in Section 4.1 are shown. A higher degree of non-linearity and deviation from the identity map corresponds to a larger decrease in calibration error. Larger uncertainty of the latent Gaussian process corresponds to less samples in the calibration data set with logits in that range.

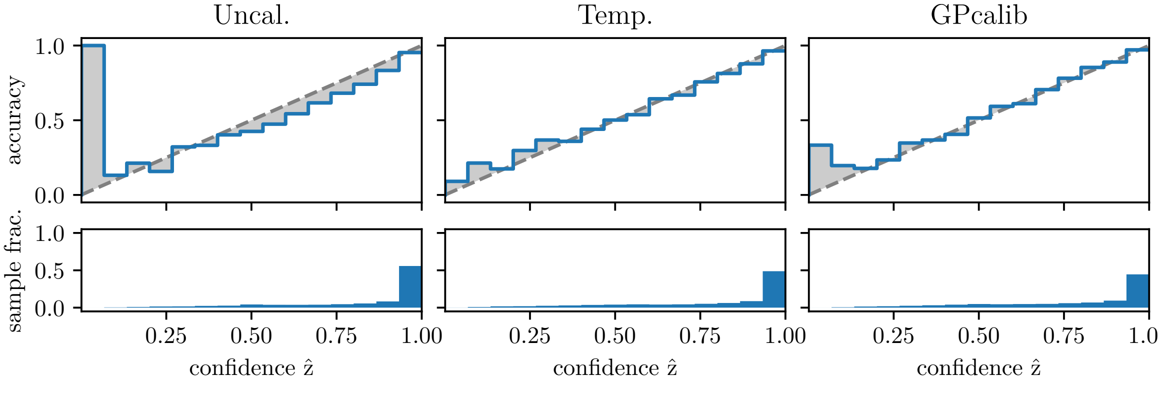

Figure S3: *Reliability diagrams of ResNet-152.* ResNet-152 is overconfident for most of its output and thus miscalibrated. Note, that the large deviation for the left-most bin is an artifact of the low number of samples in that bin and thus not representative of the true accuracy for low confidence predictions. Both temperature scaling and GPcalib improve the classifier's calibration by shifting its confidence estimates closer to the actual accuracy.

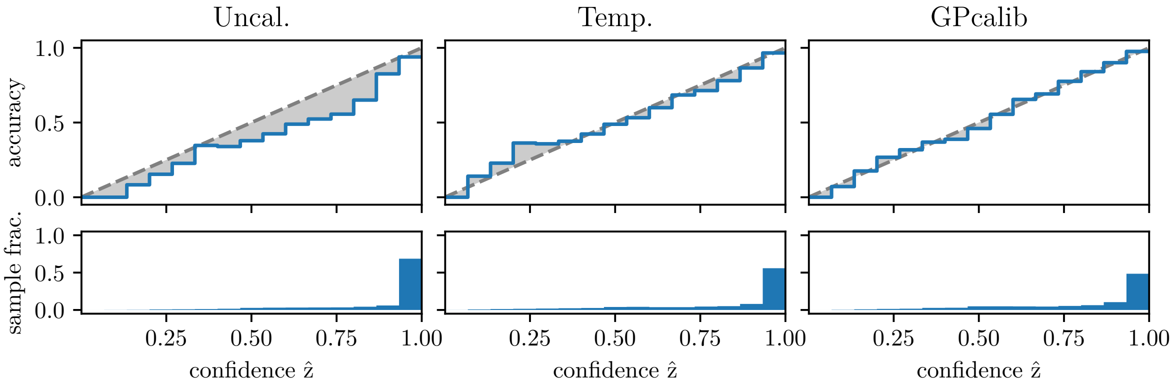

Figure S4: *Reliability diagrams of PolyNet.* PolyNet shows a similar reliability curve as Resnet-152 in Figure S3. It also displays less accuracy per bin for the given confidence in its prediction. The shown calibration methods remedy this by reducing the confidence in some of its most certain estimates.

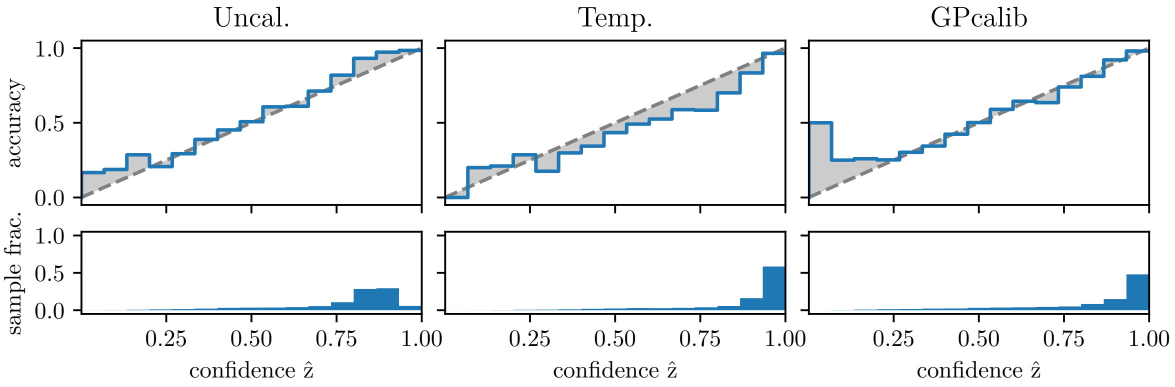

Figure S5: *Reliability diagrams of PNASNet-5-Large.* Contrary to most other CNNs, PNASNet-5-Large is actually less confident than accurate for much of its confidence histogram. We observed a similar phenomenon for other high accuracy CNNs on ImageNet in our experiments. Both calibration methods shift the histogram to more confident estimates, but GPcalib does not overcompensate for less confident predictions in contrast to temperature scaling.

Table S2: *Accuracy of binary calibration experiments.* Average accuracy and standard deviation of 10 Monte-Carlo cross validation folds on binary benchmark data sets.

Data Set	Model	Uncal.	Platt	Isotonic	Beta	BBQ	Temp.	GPcalib	GPcalib mean appr.
KITTI	AdaBoost	.9463	.9499 ± .0009	.9497 ± .0006	.9499 ± .0009	.9444 ± .0072	.9463 ± .0006	.9465 ± .0005	.9463 ± .0006
KITTI	XGBoost	.9674	.9673 ± .0007	.9660 ± .0017	.9671 ± .0011	.9640 ± .0043	.9674 ± .0006	.9675 ± .0006	.9674 ± .0006
KITTI	Mondrian Forest	.9536	.9539 ± .0004	.9523 ± .0021	.9532 ± .0009	.9439 ± .0041	.9536 ± .0003	.9536 ± .0004	.9536 ± .0003
KITTI	Random Forest	.9639	.9628 ± .0011	.9616 ± .0017	.9625 ± .0017	.8922 ± .0055	.9639 ± .0007	.9637 ± .0007	.9639 ± .0007
KITTI	1 layer NN	.9620	.9644 ± .0007	.9686 ± .0012	.9684 ± .0009	.9647 ± .0060	.9620 ± .0006	.9620 ± .0006	.9620 ± .0006
PCam	AdaBoost	.7586	.7609 ± .0030	.7644 ± .0025	.7610 ± .0030	.7638 ± .0032	.7586 ± .0022	.7588 ± .0020	.7586 ± .0022
PCam	XGBoost	.8086	.8065 ± .0013	.8050 ± .0018	.8068 ± .0015	.8020 ± .0066	.8086 ± .0016	.8084 ± .0016	.8086 ± .0016
PCam	Mondrian Forest	.7946	.7976 ± .0013	.7954 ± .0032	.7976 ± .0017	.7950 ± .0027	.7946 ± .0012	.7946 ± .0013	.7946 ± .0012
PCam	Random Forest	.8487	.8484 ± .0015	.8473 ± .0016	.8482 ± .0016	.8110 ± .0041	.8487 ± .0007	.8483 ± .0008	.8487 ± .0007
PCam	1 layer NN	.5925	.6239 ± .0070	.6504 ± .0019	.6487 ± .0031	.6458 ± .0082	.5925 ± .0008	.5779 ± .0041	.5768 ± .0033

Table S3: *Binary calibration experiments.* Average ECE_1 and standard deviation of 10 Monte-Carlo cross validation folds on binary benchmark data sets. Calibration errors (ECE_1) within one standard deviation of lowest per data set and model are printed in bold.

Data Set	Model	Uncal.	Platt	Isotonic	Beta	BBQ	Temp.	GPcalib	GPcalib mean appr.
KITTI	AdaBoost	.4301	.0182 ± .0018	**.0134** ± .0021	.0180 ± .0016	.0190 ± .0055	.0185 ± .0017	.0192 ± .0018	.0187 ± .0014
KITTI	XGBoost	.0434	.0198 ± .0019	**.0114** ± .0026	.0178 ± .0015	.0184 ± .0038	.0204 ± .0009	.0186 ± .0017	.0181 ± .0018
KITTI	Mondrian Forest	.0546	.0198 ± .0011	**.0142** ± .0018	.0252 ± .0099	.0218 ± .0035	.0200 ± .0008	.0202 ± .0018	.0203 ± .0019
KITTI	Random Forest	.0768	.0147 ± .0027	**.0135** ± .0030	.0159 ± .0027	.0652 ± .0469	**.0126** ± .0020	.0182 ± .0032	**.0135** ± .0026
KITTI	1 layer NN	.0153	.0285 ± .0034	**.0121** ± .0043	.0174 ± .0026	.0178 ± .0056	.0280 ± .0015	**.0156** ± .0020	**.0157** ± .0020
PCam	AdaBoost	.2506	.0409 ± .0020	**.0335** ± .0047	**.0397** ± .0024	**.0330** ± .0077	**.0381** ± .0033	**.0389** ± .0032	**.0388** ± .0027
PCam	XGBoost	.0605	**.0378** ± .0010	**.0323** ± .0058	**.0356** ± .0028	**.0312** ± .0110	**.0399** ± .0020	**.0341** ± .0019	**.0354** ± .0026
PCam	Mondrian Forest	.0415	.0428 ± .0024	**.0291** ± .0066	**.0349** ± .0040	.0643 ± .0161	.0427 ± .0013	**.0352** ± .0040	**.0344** ± .0040
PCam	Random Forest	.0798	.0237 ± .0035	.0233 ± .0052	.0293 ± .0053	.0599 ± .0084	**.0210** ± .0013	.0283 ± .0027	**.0212** ± .0020
PCam	1 layer NN	.2090	.0717 ± .0051	**.0297** ± .0092	.0501 ± .0049	**.0296** ± .0102	.0542 ± .0015	.0454 ± .0031	.0487 ± .0032

Table S4: *Accuracy of multi-class calibration experiments.* Average accuracy and standard deviation of 10 Monte-Carlo cross validation folds on multi-class benchmark data sets.

			one-vs-all						
Data Set	Model	Uncal.	Platt	Isotonic	Beta	BBQ	Temp.	GPcalib	GPcalib mean appr.
MNIST	AdaBoost	.7311	.6601 ± .0097	.6787 ± .0049	.6642 ± .0090	.6540 ± .0061	.7311 ± .0009	.7289 ± .0020	.7285 ± .0034
MNIST	XGBoost	.9333	.9330 ± .0011	.9312 ± .0014	.9331 ± .0011	.9274 ± .0022	.9333 ± .0006	.9333 ± .0006	.9333 ± .0006
MNIST	Mondrian Forest	.9133	.9144 ± .0015	.9118 ± .0014	.9142 ± .0015	.7475 ± .0138	.9133 ± .0008	.9132 ± .0008	.9152 ± .0009
MNIST	Random Forest	.9448	.9461 ± .0012	.9445 ± .0010	.9453 ± .0010	.0004 ± .0003	.9448 ± .0006	.9457 ± .0011	.9658 ± .0007
MNIST	1 layer NN	.9625	.9624 ± .0007	.9620 ± .0011	.9626 ± .0011	.9557 ± .0026	.9625 ± .0007	.9517 ± .0011	.9742 ± .0020
CIFAR-100	AlexNet	.4390	.4380 ± .0040	.4224 ± .0040	.4313 ± .0038	.3112 ± .0119	.4390 ± .0019	.4389 ± .0021	.4390 ± .0019
CIFAR-100	WideResNet	.8183	.8105 ± .0019	.8048 ± .0021	.8084 ± .0019	.7822 ± .0035	.8183 ± .0014	.8181 ± .0013	.8260 ± .0014
CIFAR-100	ResNeXt-29 (8x64)	.8260	.8188 ± .0020	.8137 ± .0032	.8176 ± .0023	.7941 ± .0020	.8260 ± .0014	.8259 ± .0015	.8268 ± .0012
CIFAR-100	ResNeXt-29 (16x64)	.8268	.8181 ± .0031	.8146 ± .0027	.8181 ± .0034	.7966 ± .0043	.8268 ± .0012	.8267 ± .0012	.8274 ± .0010
CIFAR-100	DenseNet-BC-190	.8274	.8218 ± .0018	.8155 ± .0021	.8189 ± .0019	.7938 ± .0040	.8274 ± .0010	.8275 ± .0010	.8183 ± .0014
ImageNet	AlexNet	.5652	.3475 ± .0071	.3455 ± .0056	.3528 ± .0058	.1866 ± .0050	.5652 ± .0040	.5653 ± .0040	.5652 ± .0040
ImageNet	VGG19	.7237	.4473 ± .0122	.4563 ± .0125	.4554 ± .0140	.2569 ± .0082	.7237 ± .0028	.7237 ± .0028	.7237 ± .0028
ImageNet	ResNet-50	.7614	.4710 ± .0087	.4810 ± .0099	.4790 ± .0097	.2660 ± .0071	.7614 ± .0042	.7614 ± .0041	.7614 ± .0042
ImageNet	ResNet-152	.7828	.4821 ± .0077	.4931 ± .0081	.4902 ± .0078	.2779 ± .0085	.7828 ± .0022	.7830 ± .0024	.7828 ± .0022
ImageNet	DenseNet-121	.7456	.4613 ± .0093	.4711 ± .0100	.4692 ± .0093	.2445 ± .0054	.7456 ± .0032	.7456 ± .0032	.7456 ± .0032
ImageNet	DenseNet-201	.7699	.4739 ± .0077	.4833 ± .0086	.4808 ± .0077	.2574 ± .0049	.7699 ± .0036	.7699 ± .0035	.7699 ± .0036
ImageNet	InceptionV4	.8011	.4958 ± .0121	.5061 ± .0118	.5043 ± .0117	.2575 ± .0056	.8011 ± .0048	.8011 ± .0047	.8011 ± .0048
ImageNet	SE-ResNeXt-50	.7905	.4849 ± .0097	.4965 ± .0097	.4945 ± .0100	.3136 ± .0102	.7905 ± .0056	.7904 ± .0056	.7905 ± .0056
ImageNet	SE-ResNeXt-101	.8022	.4995 ± .0118	.5087 ± .0121	.5081 ± .0121	.2480 ± .0058	.8022 ± .0042	.8021 ± .0043	.8022 ± .0042
ImageNet	PolyNet	.8086	.4987 ± .0082	.5122 ± .0099	.5090 ± .0098	.3232 ± .0096	.8086 ± .0030	.8083 ± .0030	.8086 ± .0030
ImageNet	SENet-154	.8117	.5035 ± .0070	.5138 ± .0061	.5144 ± .0062	.1943 ± .0055	.8117 ± .0030	.8117 ± .0030	.8117 ± .0030
ImageNet	PNASNet-5-Large	.8289	.5146 ± .0119	.5239 ± .0125	.5242 ± .0123	.1951 ± .0085	.8289 ± .0041	.8290 ± .0041	.8289 ± .0041
ImageNet	NASNet-A-Large	.8234	.5135 ± .0099	.5239 ± .0105	.5235 ± .0103	.2009 ± .0103	.8234 ± .0026	.8233 ± .0027	.8234 ± .0026

Table S5: *Multi-class calibration experiments.* Average ECE_1 and standard deviation of 10 Monte-Carlo cross validation folds on multi-class benchmark data sets. Calibration errors (ECE_1) within one standard deviation of lowest per data set and model are printed in bold.

			one-vs-all						
Data Set	Model	Uncal.	Platt	Isotonic	Beta	BBQ	Temp.	GPcalib	GPcalib mean appr.
MNIST	AdaBoost	.6121	.2267 ± .0137	.1319 ± .0108	.2222 ± .0134	.1384 ± .0104	.1567 ± .0122	**.0414** ± .0085	**.0428** ± .0123
MNIST	XGBoost	.0740	.0449 ± .0021	**.0176** ± .0018	**.0184** ± .0014	.0207 ± .0020	.0222 ± .0015	**.0180** ± .0014	**.0182** ± .0016
MNIST	Mondrian Forest	.2163	.0357 ± .0049	.0282 ± .0021	.0383 ± .0057	.0762 ± .0111	**.0208** ± .0012	**.0213** ± .0020	**.0206** ± .0019
MNIST	Random Forest	.1178	.0273 ± .0039	.0207 ± .0042	.0259 ± .0070	.1233 ± .0005	**.0121** ± .0012	.0148 ± .0021	.0135 ± .0009
MNIST	1 layer NN	.0262	**.0126** ± .0031	.0140 ± .0017	.0168 ± .0018	.0186 ± .0027	.0195 ± .0060	.0239 ± .0023	**.0109** ± .0017
CIFAR-100	AlexNet	.2751	.0720 ± .0090	.1232 ± .0102	.0784 ± .0091	.0478 ± .0058	**.0365** ± .0024	**.0369** ± .0054	**.0377** ± .0049
CIFAR-100	WideResNet	.0664	.0838 ± .0056	.0661 ± .0040	.0539 ± .0031	.0384 ± .0046	.0444 ± .0030	.0283 ± .0027	**.0246** ± .0027
CIFAR-100	ResNeXt-29 (8x64)	.0495	.0882 ± .0040	.0599 ± .0062	.0492 ± .0054	.0392 ± .0088	.0424 ± .0015	**.0251** ± .0020	**.0266** ± .0019
CIFAR-100	ResNeXt-29 (16x64)	.0527	.0900 ± .0041	.0620 ± .0058	.0520 ± .0054	.0365 ± .0078	.0465 ± .0026	.0266 ± .0018	**.0241** ± .0021
CIFAR-100	DenseNet-BC-190	.0717	.0801 ± .0061	.0665 ± .0036	.0543 ± .0036	.0376 ± .0049	.0377 ± .0019	**.0237** ± .0021	.0273 ± .0025
ImageNet	AlexNet	**.0353**	.1132 ± .0046	.2937 ± .0094	.2290 ± .0078	.1307 ± .0093	**.0342** ± .0035	**.0357** ± .0025	**.0356** ± .0024
ImageNet	VGG19	.0377	.0965 ± .0095	.2810 ± .0131	.2416 ± .0163	.1617 ± .0110	**.0342** ± .0034	**.0364** ± .0018	**.0362** ± .0016
ImageNet	ResNet-50	.0441	.0875 ± .0072	.2724 ± .0188	.2250 ± .0122	.1635 ± .0063	**.0341** ± .0018	**.0335** ± .0054	**.0330** ± .0055
ImageNet	ResNet-152	.0545	.0879 ± .0066	.2761 ± .0348	.2201 ± .0130	.1675 ± .0064	.0323 ± .0019	**.0283** ± .0038	**.0283** ± .0032
ImageNet	DenseNet-121	.0380	.0949 ± .0061	.2682 ± .0069	.2297 ± .0139	.1512 ± .0099	**.0329** ± .0020	.0357 ± .0040	.0360 ± .0033
ImageNet	DenseNet-201	.0410	.0898 ± .0111	.2706 ± .0210	.2189 ± .0157	.1614 ± .0116	**.0324** ± .0018	.0367 ± .0058	.0362 ± .0063
ImageNet	InceptionV4	.0318	.0865 ± .0102	.2900 ± .0351	.1653 ± .0138	.1593 ± .0155	.0462 ± .0070	**.0269** ± .0031	**.0274** ± .0027
ImageNet	SE-ResNeXt-50	.0440	.0889 ± .0101	.2684 ± .0260	.1789 ± .0106	.1990 ± .0176	.0482 ± .0069	**.0279** ± .0055	**.0280** ± .0045
ImageNet	SE-ResNeXt-101	.0574	.0853 ± .0112	.2844 ± .0218	.1631 ± .0098	.1496 ± .0106	.0415 ± .0039	**.0250** ± .0020	**.0253** ± .0017
ImageNet	PolyNet	.0823	.0806 ± .0048	.2590 ± .0302	.2006 ± .0124	.1787 ± .0151	.0369 ± .0030	**.0283** ± .0041	**.0276** ± .0027
ImageNet	SENet-154	.0612	.0809 ± .0106	.3003 ± .0349	.1582 ± .0098	.1502 ± .0127	.0497 ± .0028	**.0309** ± .0030	**.0303** ± .0030
ImageNet	PNASNet-5-Large	.0702	.0796 ± .0081	.3063 ± .0221	.1430 ± .0085	.1355 ± .0101	.0486 ± .0031	**.0270** ± .0018	**.0266** ± .0019
ImageNet	NASNet-A-Large	.0530	.0826 ± .0082	.3265 ± .0300	.1437 ± .0085	.1268 ± .0097	.0516 ± .0029	**.0255** ± .0030	**.0251** ± .0026