CALIBRATION OF NEURAL NETWORKS USING SPLINES
– SUPPLEMENTARY MATERIAL –

Kartik Gupta[1,2], Amir Rahimi[1], Thalaiyasingam Ajanthan[1], Thomas Mensink[3], Cristian Sminchisescu[3], Richard Hartley[1,3]
[1]Australian National University, [2]Data61, CSIRO, [3]Google Research
{kartik.gupta,amir.rahimi,thalaiyasingam.ajanthan}@anu.edu.au
{mensink,sminchisescu,richardhartley}@google.com

Here, we first provide the proof of our main result, discuss more about top-r calibration and spline-fitting, and then turn to additional experiments.

A PROOF OF PROPOSITION 4.1

We first restate our proposition below.

Proposition A.1. *If $h(t) = P(Y = k, f_k(X) \leq s(t))$ as in (14) of the main paper where $s(t)$ is the t-th fractile score. Then $h'(t) = P(Y = k \mid f_k(X) = s(t))$, where $h'(t) = dh/dt$.*

Proof. The proof is using the fundamental relationship between the Probability Distribution Function (PDF) and the Cumulative Distribution Function (CDF) and it is provided here for completeness. Taking derivatives, we see (writing $P(k)$ instead of $P(Y = k)$):

$$\begin{aligned} h'(t) &= P(k, f_k(X) = s(t)) \,.\, s'(t) \\ &= P(k \mid f_k(X) = s(t)) \,.\, P(f_k(X) = s(t)) \,.\, s'(t) \\ &= P(k \mid f_k(X) = s(t)) \,.\, \frac{d}{dt}\big(P(f_k(X) \leq s(t))\big) \\ &= P(k \mid f_k(X) = s(t)) \,.\, \frac{d}{dt}(t) \\ &= P(k \mid f_k(X) = s(t)) \,. \end{aligned} \tag{1}$$

The proof relies on the equality $P(f_k(X) \leq s(t)) = t$. In words: $s(t)$ is the value that a fraction t of the scores are less than or equal. This equality then says: the probability that a score is less than or equal to the value that a fraction t of the scores lie below, is (obviously) equal to t.

B MORE ON TOP-r AND WITHIN-TOP-r CALIBRATION

In the main paper, definitions of top-r and within-top-r calibration are given in equations (4) and (5). Here, a few more details are given of how to calibrate the classifier f for top-r and within-top-r calibration.

The method of calibration using splines described in this paper consists of fitting a spline to the cumulative accuracy, defined as h_i in equation (11) in the main paper. For top-r classification, the method is much the same as for the classification for class k. Equation (11) is replaced by sorting the data according to the r-th top score, then defining

$$\begin{aligned} \tilde{h}_0 = h_0 &= 0 \,, \\ h_i &= h_{i-1} + \mathbf{1}(y^{(-r)} = 1)/N \,, \\ \tilde{h}_i &= \tilde{h}_{i-1} + f^{(-r)}(\mathbf{x}_i)/N \,, \end{aligned} \tag{2}$$

where $y^{(-r)}$ and $f^{(-r)}(\mathbf{x}_i)$ are defined in the main paper, equation (3). These sequences may then be used both as a metric for the correct top-r calibration and for calibration using spline-fitting as described.

For within-top-r calibration, one sorts the data according to the sum of the top r scores, namely $\sum_{s=1}^{r} f^{(-s)}(\mathbf{x}_i)$, then computes

$$\begin{aligned} \tilde{h}_0 &= h_0 = 0 \,, \\ h_i &= h_{i-1} + \mathbf{1}\Big(\sum_{s=1}^{r} y^{(-s)} = 1\Big)\Big/N \,, \\ \tilde{h}_i &= \tilde{h}_{i-1} + \sum_{s=1}^{r} f^{(-s)}(\mathbf{x}_i)/N \,, \end{aligned} \tag{3}$$

As before, this can be used as a metric, or as the starting point for within-top-r calibration by our method. Examples of this type of calibration (graphs for uncalibrated networks in fig 5 and fig 7) is given in the graphs provided in fig 6 and fig 8 for within-top-2 predictions and within-top-3 predictions respectively.

It is notable that if a classifier is calibrated in the sense of equation (1) in the main paper (also called multi-class-calibrated), then it is also calibrated for top-r and within-top-r classification.

C Least Square Spline Fitting

Least-square fitting using cubic splines is a known technique. However, details are given here for the convenience of the reader. Our primary reference is (McKinley & Levine (1998)), which we adapt to least-squares fitting. We consider the case where the knot-points are evenly spaced.

We change notation from that used in the main paper by denoting points by (x, y) instead of (u, v). Thus, given knot points $(\hat{x}_i, \hat{y}_i)_{k=1}^{K}$ one is required to fit some points $(x_i, y_i)_{i=1}^{N}$. Given a point x, the corresponding spline value is given by $y = \mathbf{a}(x)^\top \mathtt{M} \hat{\mathbf{y}}$, where $\hat{\mathbf{y}}$ is the vector of values $\hat{y}_i$. The form of the vector $\mathbf{a}(x)$ and the matrix $\mathtt{M}$ are given in the following.

The form of the matrix $\mathtt{M}$ is derived from equation (25) in McKinley & Levine (1998). Define the matrices

$$\mathtt{A} = \begin{bmatrix} 4 & 1 & & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & & \ddots & & & \\ & & & & 1 & 4 & 1 \\ & & & & & 1 & 4 \end{bmatrix} \;;\; \mathtt{B} = \frac{6}{h^2} \begin{bmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & & \ddots & & & \\ & & & & 1 & -2 & 1 \end{bmatrix} ,$$

where h is the distance between the knot points. These matrices are of dimensions $K-2 \times K-2$ and $K-2 \times K$ respectively. Finally, let $\mathtt{M}$ be the matrix

$$\mathtt{M} = \begin{bmatrix} \mathbf{0}_K{}^\top \\ \mathtt{A}^{-1}\mathtt{B} \\ \mathbf{0}_K{}^\top \\ \mathtt{I}_{K\times K} \end{bmatrix} .$$

Here, $\mathbf{0}_K$ is a vector of zeros of length K, and $\mathtt{I}_{K\times K}$ is the identity matrix. The matrix $\mathtt{M}$ has dimension $2K \times K$.

Next, let the point x lie between the knots j and $j+1$ and let $u = x - \hat{x}_j$. Then define the vector $\mathbf{v} = \mathbf{a}(x)$ by values

$$\begin{aligned} v_j &= -u^3/(6h) + u^2/2 - hu/3 \,, \\ v_{j+1} &= u^3/(6h) - hu/6 \,, \\ v_{j+K} &= -u/h + 1 \,, \\ v_{j+1+K} &= u/h \,, \end{aligned}$$

with other entries equal to 0.

Dataset	Image Size	# class	Calibration set	Test set
CIFAR-10	32×32	10	5000	10000
CIFAR-100	32×32	100	5000	10000
SVHN	32×32	10	6000	26032
ImageNet	224×224	1000	25000	25000

Table 1: *Dataset splits used for all the calibration experiments. Note, "calibration" set is used for spline fitting in our method and calibration for the baseline methods and then different methods are evaluated on "test" set.*

Dataset	Model	Uncalibrated	Temp. Scaling	Vector Scaling	MS-ODIR	Dir-ODIR	**Ours (Spline)**
CIFAR-10	Resnet-110	1.805	**0.097**	0.176	0.140	0.195	0.277
	Resnet-110-SD	1.423	0.111	0.089	0.082	**0.073**	0.104
	DenseNet-40	2.256	0.435	0.409	0.395	**0.348**	0.571
	Wide Resnet-32	1.812	0.145	**0.105**	0.124	0.139	0.537
	Lenet-5	3.545	0.832	0.831	**0.631**	0.804	0.670
CIFAR-100	Resnet-110	14.270	0.885	0.649	1.425	1.190	**0.503**
	Resnet-110-SD	12.404	0.762	1.311	2.120	1.588	**0.684**
	DenseNet-40	15.901	0.437	**0.368**	2.205	0.518	0.724
	Wide Resnet-32	14.078	**0.414**	0.548	1.915	1.099	1.017
	Lenet-5	14.713	0.787	1.249	0.643	2.682	**0.518**
ImageNet	Densenet-161	4.266	1.051	0.868	3.372	2.536	**0.408**
	Resnet-152	4.851	1.167	0.776	4.093	2.839	**0.247**
SVHN	Resnet-152-SD	0.485	0.388	0.410	0.407	0.388	**0.158**

Table 2: **Within-top-2 predictions.** *KS Error (in %) within-top-2 prediction (with lowest in bold and second lowest underlined) on various image classification datasets and models with different calibration methods. Note, for this experiment we use 14 knots for spline fitting.*

Then the value of the spline is given by

$$y = \mathbf{a}(x)^\top \mathtt{M}\hat{\mathbf{y}} \ ,$$

as required. This allows us to fit the spline (varying the values of $\hat{\mathbf{y}}$) to points (x_i, y_i) by least-squares fit, as described in the main paper.

The above description is for so-called *natural* (linear-runout) splines. For quadratic-runout or cubic-runout splines the only difference is that the first and last rows of matrix $\mathtt{A}$ are changed – see McKinley & Levine (1998) for details.

As described in the main paper, it is also possible to add linear constraints to this least-squares problem, such as constraints on derivatives of the spline. This results in a linearly-constrained quadratic programming problem.

D Additional Experiments

We first provide the experimental setup for different datasets in Table 1. Note, the calibration set is used for spline fitting in our method and then final evaluation is based on an unseen test set.

We also provide comparisons of our method against baseline methods for within-top-2 predictions (equation 5 of the main paper) in Table 2 using KS error. Our method achieves comparable or better results for within-top-2 predictions. It should be noted that the scores for top-3 ($f^{(-3)}(\mathbf{x})$) or even top-4, top-5, etc., are very close to zero for majority of the samples (due to overconfidence of top-1 predictions). Therefore the calibration error for top-r with $r > 2$ predictions is very close to zero and comparing different methods with respect to it is of little value. Furthermore, for visual illustration, we provide calibration graphs of top-2 predictions in fig 1 and fig 2 for uncalibrated and calibrated network respectively. Similar graphs for top-3, within-top-2, and within-top-3 predictions are presented in figures 3 – 8.

We also provide classification accuracy comparisons for different post-hoc calibration methods against our method if we apply calibration for all top-$1, 2, 3, \ldots, K$ predictions for K-class classification problem in Table 3. We would like to point out that there is negligible change in accuracy between the calibrated networks (using our method) and the uncalibrated ones.

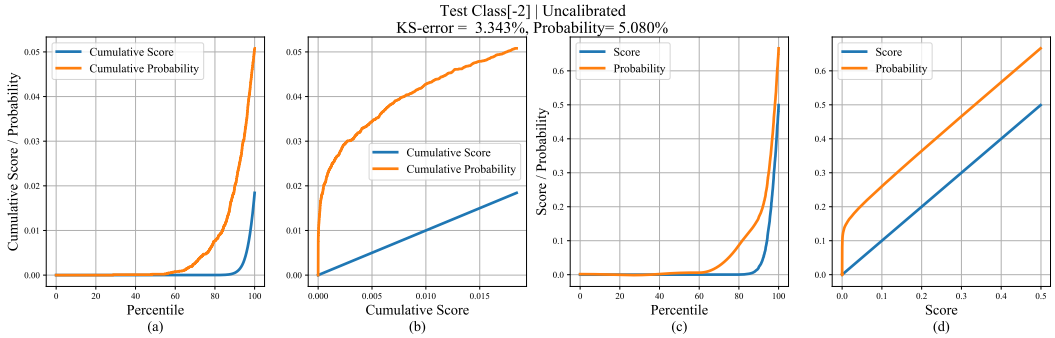

Figure 1: **Top-2 predictions, Uncalibrated.** *Calibration graphs for an uncalibrated DenseNet-40 (Huang et al. (2017)) trained on CIFAR-10 for top-2 class with a KS error of* 3.343% *on the test set. Here (**a**) shows the plot of cumulative score and probability versus the fractile of the test set, (**b**) shows the same information with the horizontal axis warped so that the cumulative-score graph is a straight line. This is created as scatter plots of cumulative (score, score): blue and (score, probability): orange. If the network is perfectly calibrated, the probability line will be a straight line coincident with the (score, score) line. This shows that the network is substantially overestimating (score) the probability of the computation. (**c**) and (**d**) show plots of (non-cumulative) score and probability plotted against fractile, or score. How these plots are produced is described in Section 4 of main paper.*

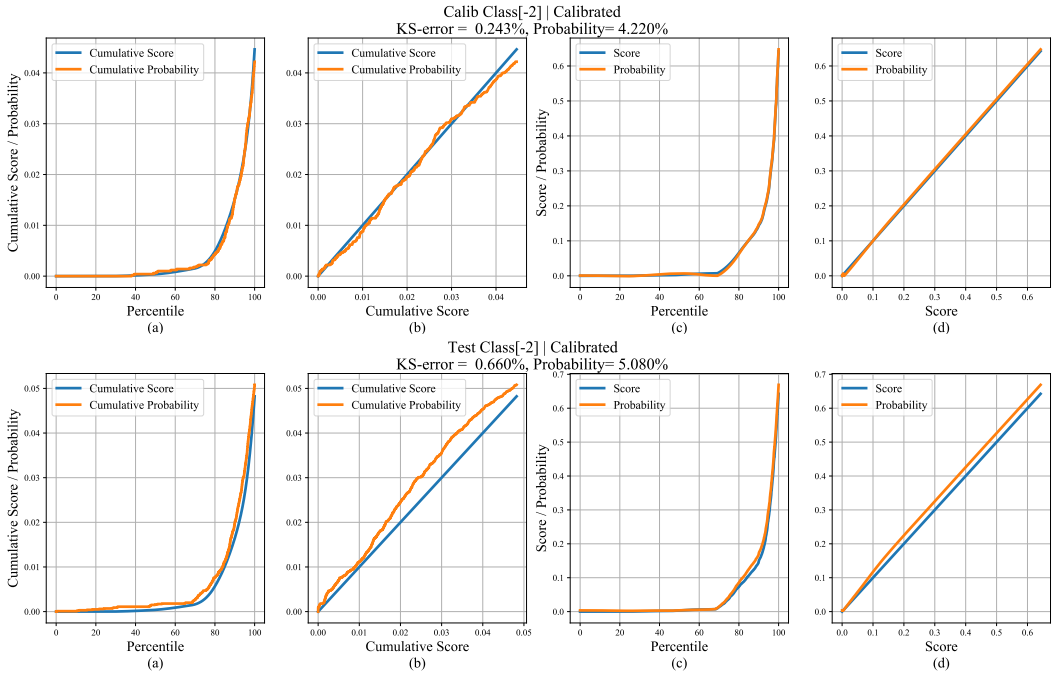

Figure 2: **Top-2 predictions, Calibrated.** *The result of the spline calibration method, on the example given in fig 1 for top-2 calibration. A recalibration function $\gamma : \mathbb{R} \to \mathbb{R}$ is used to adjust the scores, replacing $f_k(\mathbf{x})$ with $\gamma(f_k(\mathbf{x}))$ (see Section 4 of main paper). As is seen, the network is now almost perfectly calibrated when tested on the "calibration" set (**top row**) used to calibrate it. In **bottom row**, the recalibration function is tested on a further set "test". It is seen that the result is not perfect, but much better than the original results in fig 1d.*

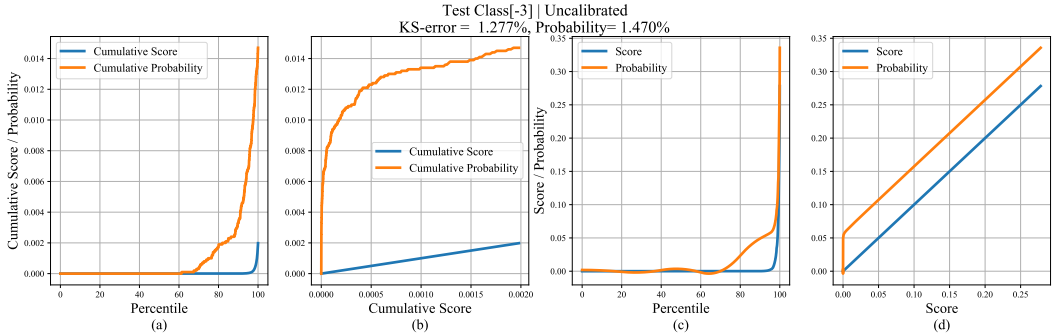

Figure 3: **Top-3 predictions, Uncalibrated.** *Calibration graphs for an uncalibrated DenseNet-40 trained on CIFAR-10 for top-3 class with a KS error of* 1.277% *on the test set. Here (**a**) shows the plot of cumulative score and probability versus the fractile of the test set, (**b**) shows the same information with the horizontal axis warped so that the cumulative-score graph is a straight line. (**c**) and (**d**) show plots of (non-cumulative) score and probability plotted against fractile, or score.*

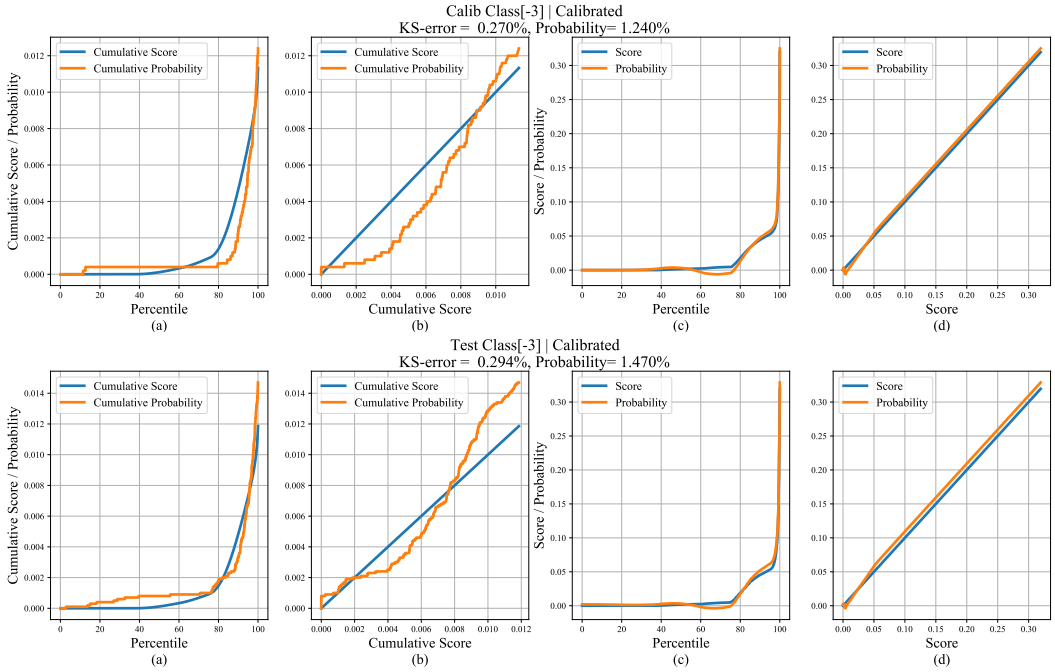

Figure 4: **Top-3 predictions, Calibrated.** *The result of the spline calibration method, on the example given in fig 3 for top-3 calibration. A recalibration function* $\gamma : \mathbb{R} \to \mathbb{R}$ *is used to adjust the scores, replacing* $f_k(\mathbf{x})$ *with* $\gamma(f_k(\mathbf{x}))$*. As is seen, the network is now almost perfectly calibrated when tested on the "calibration" set (**top row**) used to calibrate it. In **bottom row**, the recalibration function is tested on a further set "test". It is seen that the result is not perfect, but much better than the original results in fig 3d.*

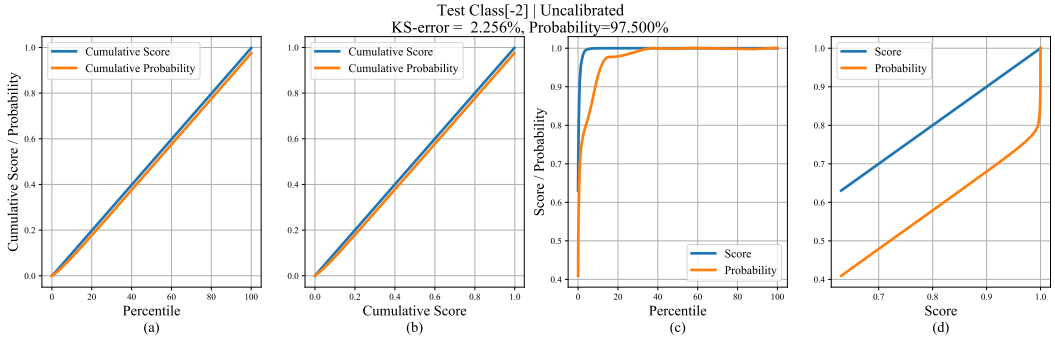

Figure 5: **Within-top-2 predictions, Uncalibrated.** *Calibration graphs for an uncalibrated DenseNet-40 trained on CIFAR-10 for within-top-2 predictions with a KS error of* 2.256% *on the test set. Here (**a**) shows the plot of cumulative score and probability versus the fractile of the test set, (**b**) shows the same information with the horizontal axis warped so that the cumulative-score graph is a straight line. (**c**) and (**d**) show plots of (non-cumulative) score and probability plotted against fractile, or score.*

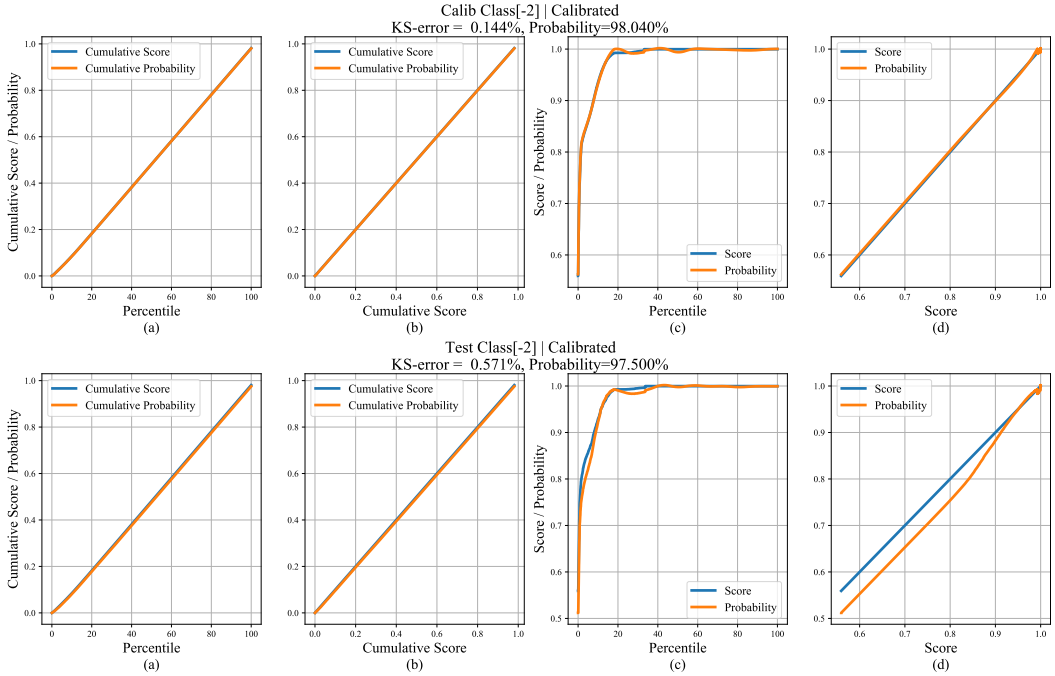

Figure 6: **Within-top-2 predictions, Calibrated.** *The result of the spline calibration method, on the example given in fig 5 for within-top-*2 *calibration. A recalibration function* $\gamma : \mathbb{R} \to \mathbb{R}$ *is used to adjust the scores, replacing* $f_k(\mathbf{x})$ *with* $\gamma(f_k(\mathbf{x}))$*. As is seen, the network is now almost perfectly calibrated when tested on the "calibration" set (**top row**) used to calibrate it. In **bottom row**, the recalibration function is tested on a further set "test". It is seen that the result is not perfect, but much better than the original results in fig 5d.*

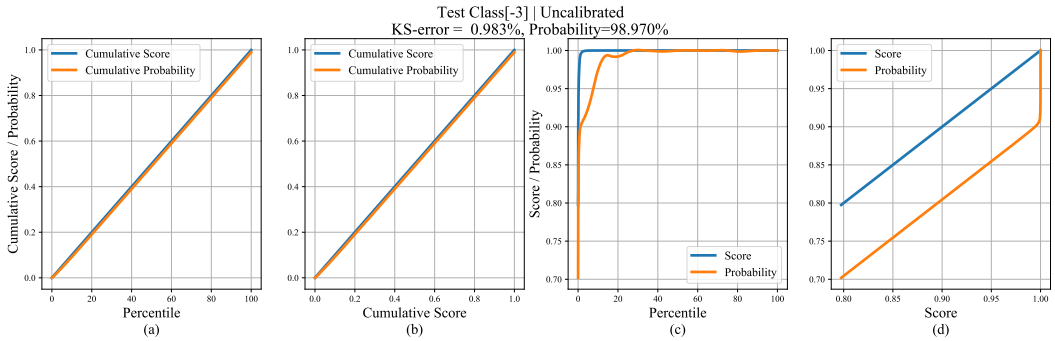

Figure 7: **Within-top-3 predictions, Uncalibrated.** *Calibration graphs for an uncalibrated DenseNet-40 trained on CIFAR-10 for within-top-3 predictions with a KS error of* 0.983% *on the test set. Here (**a**) shows the plot of cumulative score and probability versus the fractile of the test set, (**b**) shows the same information with the horizontal axis warped so that the cumulative-score graph is a straight line. (**c**) and (**d**) show plots of (non-cumulative) score and probability plotted against fractile, or score.*

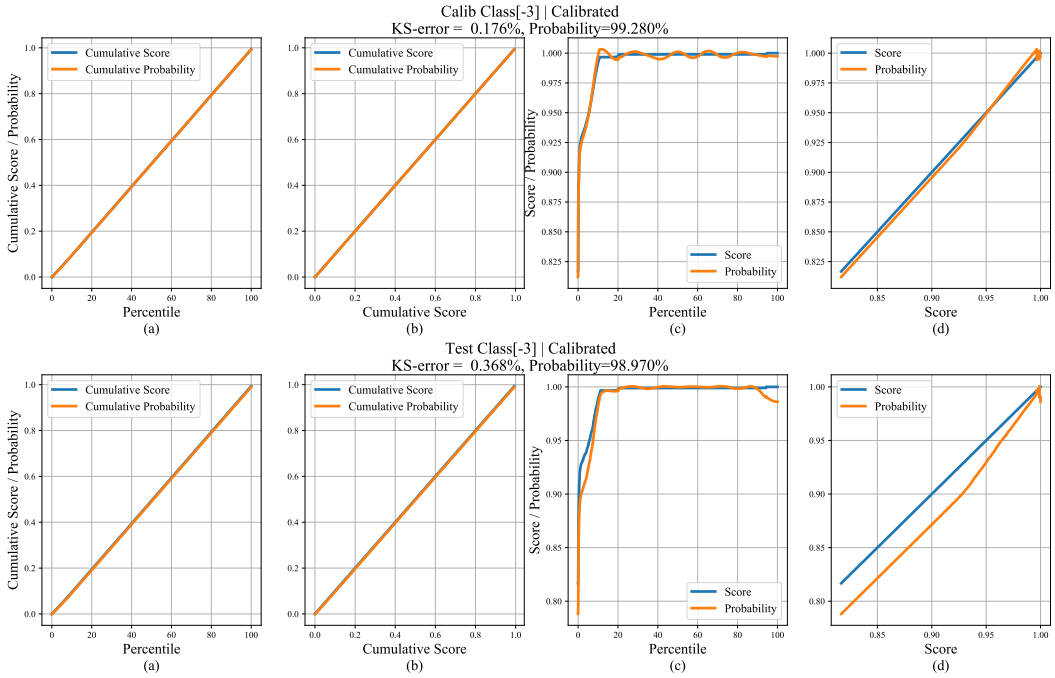

Figure 8: **Within-top-3 predictions, Calibrated.** *The result of the spline calibration method, on the example given in fig 7 for within-top-3 calibration. A recalibration function* $\gamma : \mathbb{R} \to \mathbb{R}$ *is used to adjust the scores, replacing* $f_k(\mathbf{x})$ *with* $\gamma(f_k(\mathbf{x}))$*. As is seen, the network is now almost perfectly calibrated when tested on the "calibration" set (**top row**) used to calibrate it. In **bottom row**, the recalibration function is tested on a further set "test". It is seen that the result is not perfect, but much better than the original results in fig 7d.*

Dataset	Model	Uncalibrated	Temp. Scaling	Vector Scaling	MS-ODIR	Dir-ODIR	**Ours (Spline)**
CIFAR-10	Resnet-110	**93.56**	**93.56**	93.50	93.53	93.52	93.55
	Resnet-110-SD	94.04	94.04	94.04	94.18	**94.20**	94.05
	DenseNet-40	92.42	92.42	92.50	**92.52**	92.47	92.31
	Wide Resnet-32	93.93	93.93	94.21	**94.22**	**94.22**	93.76
	Lenet-5	72.74	72.74	74.48	74.44	**74.52**	72.64
CIFAR-100	Resnet-110	71.48	71.48	71.58	71.55	**71.62**	71.50
	Resnet-110-SD	72.83	72.83	**73.60**	73.53	73.14	72.81
	DenseNet-40	70.00	70.00	70.13	**70.40**	70.24	70.17
	Wide Resnet-32	73.82	73.82	73.87	**74.05**	73.99	73.74
	Lenet-5	33.59	33.59	36.42	**37.58**	37.52	33.55
ImageNet	Densenet-161	77.05	77.05	76.72	77.15	**77.19**	77.05
	Resnet-152	76.20	76.20	75.87	76.12	**76.24**	76.07
SVHN	Resnet-152-SD	98.15	98.15	98.13	98.12	**98.19**	98.17

Table 3: *Classification (top-1) accuracy (with highest in bold and second highest underlined) post calibration on various image classification datasets and models with different calibration methods. Note, only a negligible change in accuracy is observed in our method compared to the uncalibrated networks.*

Dataset	Model	Uncalibrated	Temp. Scaling	Vector Scaling	MS-ODIR	Dir-ODIR	**Ours (Spline)**
CIFAR-10	Resnet-110	4.750	1.224	1.092	1.276	1.240	**1.011**
	Resnet-110-SD	4.135	0.777	0.752	**0.684**	0.859	0.992
	DenseNet-40	5.507	**1.006**	1.207	1.250	1.268	1.389
	Wide Resnet-32	4.512	0.905	**0.852**	0.941	0.965	1.003
	Lenet-5	5.188	1.999	1.462	1.504	**1.300**	1.333
CIFAR-100	Resnet-110	18.480	2.428	2.722	3.011	2.806	**1.868**
	Resnet-110-SD	15.861	**1.335**	2.067	2.277	2.046	1.766
	DenseNet-40	21.159	**1.255**	1.598	2.855	1.410	2.114
	Wide Resnet-32	18.784	**1.667**	1.785	2.870	2.128	1.672
	Lenet-5	12.117	1.535	1.350	1.696	2.159	**1.029**
ImageNet	Densenet-161	5.720	2.059	2.637	4.337	3.989	**0.798**
	Resnet-152	6.545	2.166	2.641	5.377	4.556	**0.913**
SVHN	Resnet-152-SD	0.877	0.675	**0.630**	0.646	0.651	0.832

Table 4: *ECE for top-1 predictions (in %) using 25 bins (with lowest in bold and second lowest underlined) on various image classification datasets and models with different calibration methods. Note, for this experiment we use 13 knots for spline fitting.*

For the sake of completeness, we present calibration results using the existing calibration metric, Expected Calibration Error (ECE) (Naeini et al. (2015)) in Table 4. We would like to reiterate the fact that ECE metric is highly dependent on the chosen number of bins and thus does not really reflect true calibration performance. To reflect the efficacy of our proposed calibration method, we also present calibration results using other calibration metrics such as recently proposed binning free measure KDE-ECE (Zhang et al. (2020)), MCE (Maximum Calibration Error) (Guo et al. (2017)) and Brier Scores for top-1 predictions on ImageNet dataset in Table 5. Since, the original formulation of Brier Score for multi-class predictions is highly biased on the accuracy and is approximately similar for all calibration methods, we hereby use top-1 Brier Score which is the mean squared error between top-1 scores and ground truths for the top-1 predictions (1 if the prediction is correct and 0 otherwise). It can be clearly observed that our approach consistently outperforms all the baselines on different calibration measures.

REFERENCES

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1321–1330. JMLR. org, 2017.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Sky McKinley and Megan Levine. Cubic spline interpolation. *College of the Redwoods*, 1998.

Calibration Metric	Model	Uncalibrated	Temp. Scaling	MS-ODIR	Dir-ODIR	**Ours (Spline)**
KDE-ECE	Densenet-161	0.03786	<u>0.01501</u>	0.02874	0.02979	**0.00637**
	Resnet-152	0.04650	<u>0.01864</u>	0.03448	0.03488	**0.00847**
MCE	Densenet-161	0.13123	**0.05442**	0.09077	0.09653	<u>0.06289</u>
	Resnet-152	0.15930	<u>0.09051</u>	0.11201	0.09868	**0.04950**
Brier Score	Densenet-161	0.12172	<u>0.11852</u>	0.11982	0.11978	**0.11734**
	Resnet-152	0.12626	<u>0.12145</u>	0.12406	0.12308	**0.12034**

Table 5: *Calibration Error using other different metrics such as binning-free KDE-ECE (Zhang et al. (2020)), MCE (Maximum Calibration Error) (Guo et al. (2017)) and Brier Score for top-1 predictions (with lowest in bold and and second lowest underlined) on ImageNet dataset with different calibration methods. Note, for this experiment we use 6 knots for spline fitting.*

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Jize Zhang, Bhavya Kailkhura, and T Han. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. *ICML*, 2020.