Supplementary Material

A Stitch in Time Saves Nine: A Train-Time Regularizing Loss for Improved Neural Network Calibration

Table of Contents

Due to the restrictions on the length, we could not include descriptions of the state-of-the-art techniques, datasets, hyper-parameters and additional results in the main manuscript. However, to keep the overall manuscript self-contained, we include the following in the supplementary material:

- Section 1: Description of competing methods.
- Section 2: Detailed description of benchmark datasets used in our experiments.
- Section 3: Training procedure and implementation of proposed, and competing methods.
- Section 4: Additional results.
- Source code link: MDCA Official PyTorch Code

1. Competing Methods and hyperparameters used

In this section, we provide a brief description of each of the compared method with hyperparameter settings used in training.

- For `Brier Score` [1] we train on the loss defined as the squared loss of the predicted probability vector and the one-hot target vector.
- `Label Smoothing` [17], takes the form $\texttt{LS} = -\sum_{i=1}^{N}\sum_{j=1}^{K} q_{i,j}\log(\hat{p}_{i,j})$ where $\hat{p}_{i,j}$ is the predicted confidence score for sample i, for class j. Similarly, we define soft target vector, q_i, for each sample i, such that $q_{i,j} = \alpha/(K-1)$ if $j \neq y_i$, else $q_{i,j} = (1-\alpha)$. Here α is a hyper-parameter. We trained using $\alpha = 0.1$, and refer to label smoothing as `LS` in the results.
- `Focal loss` [15] is defined as $\texttt{FL} = -\sum_{i=1}^{N}(1-\hat{p}_{i,y_i})^{\gamma}\log(\hat{p}_{i,y_i})$, where γ is a hyper-parameter. We trained using $\gamma \in \{1,2,3\}$, and report it as `FL` in the results.
- For DCA [13], we train on the following loss: $\texttt{NLL} + \beta \cdot \texttt{DCA}$, where `DCA` is as defined in [13], and β is a hyper-parameter. We train varying $\beta \in \{1,5,10,15,20,25\}$ as performed in [13]. DCA results are reported under the name `DCA`.
- We use `MMCE` [10], as a regularizer along with `NLL`. We use the weighted `MMCE` loss in our experiments with $\lambda \in \{2,4\}$.
- For `FLSD` [16], we train with $\gamma = 3$.

For each of the above methods, we report the result of the best performing trained model according to the accuracy obtained on the validation set.

2. Dataset description

We have used the following datasets in our experiments:

1. **CIFAR10** [8]: This dataset has $60,000$ color images of size 32×32 each, equally divided into 10 classes. The pre-divided train set comes with $50,000$ images and the test set has around $10,000$ images. Using the policy defined above, we have a train/val/test split having $45000/5000/10000$ images respectively.
2. **CIFAR100** [8]: This dataset comprises of $60,000$ colour images of size $32x32$ each, but this time, equally divided into 100 classes. The pre-divided train set again comes with $50,000$ images and the test set has around $10,000$ images. We have a train/val/test split having $45000/5000/10000$ images respectively.
3. **SVHN** [18] : The Street View House Number (SVHN) is a digit classification benchmark dataset that contains 600000 32×32 RGB images of printed digits (from 0 to 9) cropped from pictures of house number plates. The cropped images are centered in the digit of interest,

but nearby digits and other distractors are kept in the image. SVHN has comes with a training set (73257) and a testing set (26032). We randomly sample 10% of the training set to use as a validation set.

4. **Tiny-ImageNet** [4] : It is a subset of the ImageNet dataset containing 64×64 RGB images. It has 200 classes with each class having 500 images. The validation set contains 50 images per class. We use the provided validation set as the test set for our experiments.

5. **20 Newsgroups** [11]: It is a popular text classification dataset containing $20,000$ news articles, categorised evenly into 20 different newsgroups based on their content.

6. **Mendeley V2** [7]: Inspired from [13], we use this medical dataset. The dataset contains OCT (optical coherence tomography) images of retina and pediatric chest X-ray images. However, we only use the chest X-ray images in our experiments. The chest X-ray images come with a pre-defined train/test split having 4273 pneumonia images and 1583 normal images of the chest.

7. **PACS dataset** [12]: We use the dataset to study calibration under domain shift. The dataset comprises of a total 9991 images spread across 4 different domains with 7 classes each. The domains are namely Photo, Art, Sketch and Cartoon. We fine-tune the ResNet-18 model, pretrained ImageNet dataset, on the Photo domain using various competing techniques and test on the other three domains to measure how calibration holds in a domain shift. Following [12], we also divide the training set of photo domain into $9:1$ train/val set.

8. **Rotated MNIST Dataset**: This dataset is also used for domain shift experiments. Inspired from [21], we create 5 different test sets namely $\{M_{15}, M_{30}, M_{45}, M_{60}, M_{75}\}$. Domain drift is introduced in each M_x by rotating the images in the MNIST test set by x degrees counter-clockwise.

9. **Segmentation Datasets- PASCAL VOC 2012 [5]**: This a segmentation dataset and consists of images with pixel-level annotations. There are 21 classes overall (One background class and 20 foreground object classes). The dataset is divided into *Train* (1464 images), *Val* (1449 images) and *Test* (1456 images) set. We, however, only make use of the *Train* and the *Val* set. Models are trained on the *Train* set and the evaluation is reported on the *Val* set.

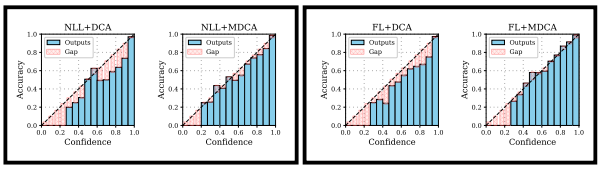

Figure 1. Comparison of Reliability diagrams of: (left) `NLL+DCA` vs `NLL+MDCA` and (right) `FL+DCA` vs. `FL+MDCA`. We use ResNet-32 trained on CIFAR10 dataset for comparison.

3. Training Procedure and Implementation

Backbone Architecture: We use ResNet [6] backbone for most of our experiments. For training on CIFAR10, and CIFAR100 datasets we used ResNet-32 and ResNet-56. For SVHN we used ResNet-20 and ResNet-56. Following [13], for Mendeley V2 dataset, we use a ResNet-50 architecture pre-trained on ImageNet dataset [4]. We use the backbone as a fixed feature extractor and add a 1×1 convolutional layer and two fully connected layers on top of the feature extractor. Segmentation experiments make use of DeepLabV3+ [2] based on the Xception65 [3] backbone

Train Parameters: For all our experiments we used a single Nvidia 1080 Ti GPU. We trained CIFAR10 for a total of 160 epochs using a learning rate of 0.1, and it is reduced by a factor of 10 at the 80^{th} and 160^{th} epoch. Train batch size was kept at 128 and DNN was optimized using Stochastic Gradient Descent (SGD) with momentum at 0.9 and weight decay being 0.0005. Furthermore, we augment the images using random center crop and horizontal flips. We use same parameters for CIFAR100, except the number of epochs, where we train it for 200 epochs. Again learning rate is reduced by a factor of 10, but this time it is reduced at epochs 100 and 150. For Tiny-ImageNet, we followed the same training procedure as done by [16]. For SVHN, we keep the same training procedure as above except the number of epochs; we train for 100 epochs, with it getting reduced at epochs 50 and 70 with a factor of 10 yet again. We do not augment the images when training on SVHN. We use PyTorch framework for all our implementations. Our repository is inspired from `https://github.com/bearpaw/pytorch-classification`. We also take help from the official implementation of [16] to implement some of the baseline methods. For the segmentation experiments we make use of the following repository: `https://github.com/LikeLy-Journey/SegmenTron`. We train the segmentation models for 120 epochs using SGD optimizer with a warm-up LR scheduler. To train with focal loss, we used $\gamma = 3$. The rest of the parameters for the optimizer and the scheduler were kept the same as provided by the SegmenTron repository.

Optimizing Hyper-parameter β for MDCA: We vary the hyper-parameter $\beta \in [0.25, 0.5, 1, 5, 10, \dots 50]$ in our exper-

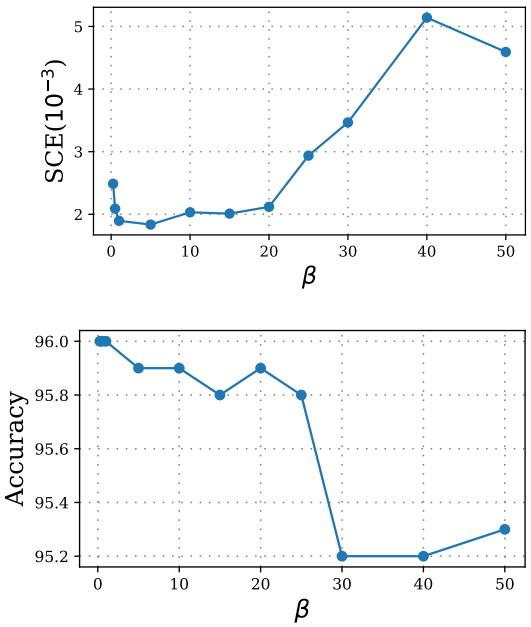

Figure 2. Effect of β on Accuracy and SCE (10^{-3}) on FL+MDCA on the ResNet-20 model trained on SVHN dataset.

$\mathcal{L}_C$	$\mathcal{L}_{Auxiliary}$	Accuracy	SCE (10^{-3})	ECE (%)
ResNet-32 on CIFAR10 dataset:				
Cross Entropy	None	92.54	8.68	4.25
	DCA [13]	92.94	8.41	4.00
	MMCE [10]	91.59	8.17	3.31
	MDCA (ours)	91.85	4.63	1.69
Label Smoothing [17]	None	92.48	14.09	6.28
	DCA [13]	88.69	8.14	3.44
	MMCE [10]	91.92	22.49	10.07
	MDCA (ours)	93.01	12.38	5.66
Focal Loss [15]	None	92.52	4.09	1.01
	DCA [13]	92.65	7.50	3.49
	MMCE [10]	90.76	26.41	13.38
	MDCA (ours)	92.82	3.22	0.93
ResNet-20 on SVHN dataset:				
Cross Entropy	None	96.14	3.43	1.64
	DCA [13]	96.17	4.29	2.02
	MMCE [10]	95.88	9.18	4.34
	MDCA (ours)	96.33	1.46	0.43
Label Smoothing [17]	None	96.24	18.80	8.88
	DCA [13]	96.55	9.80	4.46
	MMCE [10]	96.11	31.75	15.50
	MDCA (ours)	96.14	13.91	6.46
Focal Loss [15]	None	96.17	2.54	0.89
	DCA [13]	96.35	2.12	0.44
	MMCE [10]	95.83	25.01	12.79
	MDCA (ours)	96.08	1.90	0.47

Table 1. Ablation to study the impact of various auxiliary losses on two models : ResNet-32 on CIFAR10 and Resnet-20 trained model on SVHN Dataset. Our approach exhibits least calibration error without sacrificing on the accuracy.

iments. Figure 2 shows how calibration error and accuracy is affected when we increase β in different model-dataset pairs. We see a general trend that calibration is best achieved when β is close to 1, and as we increase it, the calibration as well as the accuracy starts to drop (accuracy decreases and the SCE score increases).

Post-Hoc Calibration Experiments: For comparison with post-hoc techniques, we set aside 10% of the training data as a validation set (hold-out set) to conduct post-hoc calibration. For Temperature Scaling (TS), we perform a grid search between the range of 0 to 10 with a step-size of 0.1 to find the optimal temperature value that gives the least NLL on the hold-out set. For Dirichlet Calibration (DC), we attach a single layer neural network at the end of the DNN and use ODIR [9] regularization on the weights of the same. We train on the hold-out set keeping the rest of the DNN weights frozen except the newly added layer. We again use grid search to find the optimal hyper-parameters λ and μ that give the least NLL on the hold-out set. We vary $\lambda \in \{0, 0.01, 0.1, 1, 10, 0.005, 0.05, 0.5, 5, 0.0025, 0.025, 0.25, 2.5\}$ and $\mu \in \{0, 0.01, 0.1, 1, 10\}$.

Domain Shift Experiments: For PACS dataset, we use the official PyTorch ResNet-18 model pre-trained on ImageNet dataset. We re-initialized its last fully connected layer to accommodate 7 classes, and finally fine-tuned on the Photo domain. We use the SGD optimizer with same momentum and weight decay values as done for CIFAR10/100 and described earlier. Training batch size was fixed at 256, and the model was trained for 30 epochs with initial learning rate, set at 0.01, getting reduced at epoch 20 with a factor of 10. Training parameters are chosen such that they give the best performing model i.e. having the maximum accuracy on the Photo domain val set.

For Rotated MNIST, we used PyTorch framework to generate rotated images. We use the ResNet-20 model to train on the standard MNIST train set for 30 epochs with a learning rate of 0.1 for first 20 epochs, and then with 0.01 for the last 10 epochs. Rest of the details like batch size and optimizer remain same as the CIFAR10/100 experiments. We did not augment any images, and selected the training parameters such that the model gives best accuracy on the validation set.

For 20 Newsgroups, we train the Global Pooling Convolutional Network [14] using the ADAM optimizer, with learning rate 0.001, and the default values of betas at 0.9 and 0.999 respectively. We used GloVe word embeddings [19] to train the network. We trained the model for 50 epochs.

4. Additional Results

We report the following additional results:

1. **Class-j-ECE score:** In Table 3 of main manuscript we reported the Class-j-ECE score for SVHN. In Table 6

Methods	Art	Cartoon	Sketch	Average
SCE(10^{-3})				
NLL	6.33	17.95	15.01	13.10
LS [17]	7.80	11.95	**10.88**	10.21
FL [15]	8.61	16.62	10.94	12.06
Brier Score [1]	6.55	13.19	15.63	11.79
MMCE [10]	6.35	15.70	17.16	13.07
DCA [13]	7.49	18.01	14.99	13.49
FLSD [16]	8.35	13.39	13.86	11.87
Ours (FL+MDCA)	**6.21**	**11.91**	11.08	**9.73**
ECE (%)				
Brier Score [1]	3.35	33.68	42.61	26.55
NLL	9.42	52.99	35.56	32.66
LS [17]	8.70	25.21	13.29	15.73
Focal Loss [15]	7.34	48.96	25.33	27.21
MMCE [10]	17.06	43.25	40.79	33.70
DCA [13]	13.38	55.20	37.76	35.45
FLSD [16]	8.41	34.43	30.01	24.28
Ours (FL+MDCA)	6.29	29.81	23.05	19.71
Top-1 (%) Accuracy				
Brier Score [1]	59.28	24.83	28.43	37.51
NLL	59.08	21.20	28.00	36.09
LS [17]	56.35	22.01	29.88	36.08
FL [15]	52.83	17.32	26.70	32.28
MMCE [10]	60.60	29.95	20.36	36.97
DCA [13]	57.91	20.86	28.30	35.69
FLSD [16]	54.54	20.82	29.35	34.90
Ours (FL+MDCA)	63.23	27.86	23.01	38.03

Table 2. Table comparing SCE, ECE and Top-1% Accuracy values of our method against others on PACS [12] dataset when the model is trained on `Photo` domain and tested on domains: `Art, Cartoon, and Sketch`

here, we we provide additional results for CIFAR10.

2. **Comparison with other auxiliary losses**: In Table 6 in the main manuscript showed how the proposed MDCA can be used along with `NLL`, `LS` [20], and `FL` [15] to improve the calibration performance without sacrificing the accuracy. In Tab. 1 here, we show a similar comparison for other competitive approaches, namely `DCA` [13], and `MMCE` [10]. Using `MDCA`, gives better calibration than other competitive approaches.

3. **Calibration performance under dataset drift:** A model trained using our proposed loss gives better calibration under dataset drift as well. Table 4 in the main manuscript showed SCE score comparison on PACS. We give more detailed comparison here in Tab. 2 which shows top 1% accuracy, ECE as well. We repeat SCE numbers from main manuscript for completion. Tab. 3 shows the corresponding numbers for Rotated MNIST.

   Just like we showed that using `MDCA` in conjunction with `NLL`, `LS` [20], and `FL` [15] gives best calibration performance, we show that this remains true even for the dataset drift case. Tab. 4 and Tab. 5 show the comparison on Rotated MNIST and PACS datasets respectively.

4. **Reliability Diagrams:** Fig. 2 in the main manuscript showed reliability and confidence plots for `MDCA` used with `NLL` and `LS` respectively. We show similar plots for `MDCA+FL` in Fig. 3.

Method	Clean	M_{15}	M_{30}	M_{45}	M_{60}	M_{75}	Average
SCE (%)							
NLL	0.07	0.19	0.70	2.96	7.50	10.60	3.67
LS [17]	2.00	2.00	1.93	2.91	6.67	8.93	4.07
FL [15]	0.29	0.81	1.34	2.62	7.04	10.10	3.70
Brier Score [1]	0.23	0.51	1.09	2.83	6.58	9.10	3.39
MMCE [10]	2.51	4.05	5.01	4.55	5.28	5.29	4.45
DCA [13]	0.07	0.20	0.91	3.71	8.42	11.65	4.16
FLSD [16]	1.30	2.09	3.10	3.05	4.88	7.56	3.67
Ours (FL+MDCA)	0.20	0.48	0.94	2.51	6.65	9.61	3.40
ECE (%)							
NLL	0.11	0.38	1.13	10.54	31.95	45.06	14.86
LS [17]	9.89	9.32	7.13	6.86	27.36	37.28	16.31
FL [15]	1.34	3.11	3.92	5.21	28.11	41.17	13.81
Brier Score [1]	0.90	2.07	2.57	5.95	25.24	38.16	12.48
MMCE [10]	4.72	15.42	23.01	18.88	10.74	6.80	13.26
DCA [13]	0.21	0.29	1.88	13.83	36.31	50.25	17.13
FLSD [16]	6.50	10.43	14.33	7.92	14.80	29.34	13.89
Ours (FL+MDCA)	0.79	2.03	2.94	6.18	25.77	39.29	12.83
Top-1 (%) Accuracy							
NLL	99.61	98.79	93.38	73.82	43.24	24.07	72.15
LS [17]	99.62	98.39	92.04	69.33	39.94	22.99	70.39
FL [15]	99.63	98.20	91.87	69.95	38.67	20.83	69.86
Brier Score [1]	99.63	98.72	92.54	71.29	41.90	22.73	71.14
MMCE [10]	98.43	94.99	83.93	53.88	28.76	16.84	62.81
DCA [13]	99.60	98.36	91.51	68.34	38.32	20.93	69.51
FLSD [16]	99.67	98.79	92.77	71.17	40.17	20.84	70.57
Ours (FL+MDCA)	99.59	98.61	93.13	70.92	41.93	23.37	71.26

Table 3. Table comparing SCE, ECE and Top-1% Accuracy values of our loss with other methods on the Rotated-MNIST dataset trained using a ResNet-20 model.

Domain	NLL	NLL+MDCA	LS	LS+MDCA	FL	FL+MDCA
SCE (%)						
Clean	0.07	0.07	2.00	1.99	0.29	0.20
M_{15}	0.19	0.19	2.00	1.99	0.81	0.48
M_{30}	0.70	0.66	1.93	1.88	1.34	0.94
M_{45}	2.96	3.17	2.91	2.85	2.62	2.51
M_{60}	7.50	7.99	6.67	6.08	7.04	6.65
M_{75}	10.60	11.17	8.93	8.82	10.10	9.61
Average	**3.67**	3.88	4.07	**3.94**	3.70	**3.40**
ECE (%)						
Clean	0.11	0.22	9.89	9.83	1.34	0.79
M_{15}	0.38	0.21	9.32	9.41	3.11	2.03
M_{30}	1.13	1.31	7.13	7.38	3.92	2.94
M_{45}	10.54	12.50	6.86	8.41	5.21	6.18
M_{60}	31.95	35.02	27.36	25.32	28.11	25.77
M_{75}	45.06	49.41	37.28	38.41	41.17	39.29
Average	14.86	16.45	16.31	16.46	13.81	12.83
Top-1 (%) Accuracy						
Clean	99.61	99.64	99.62	99.63	99.63	99.59
M_{15}	98.79	98.63	98.39	98.45	98.20	98.61
M_{30}	93.38	93.23	92.04	92.54	91.87	93.13
M_{45}	73.82	71.53	69.33	69.74	69.95	70.92
M_{60}	43.24	41.27	39.94	41.88	38.67	41.93
M_{75}	24.07	22.05	22.99	22.97	20.83	23.37
Average	72.15	71.06	70.39	70.87	69.86	71.26

Table 4. Table comparing SCE, ECE and Top-1% Accuracy values of our method against others for ResNet-20 model on Rotated-MNIST dataset. *Clean* denotes the original dataset, and subscript under the 'M' indicates angle of rotation for each digit.

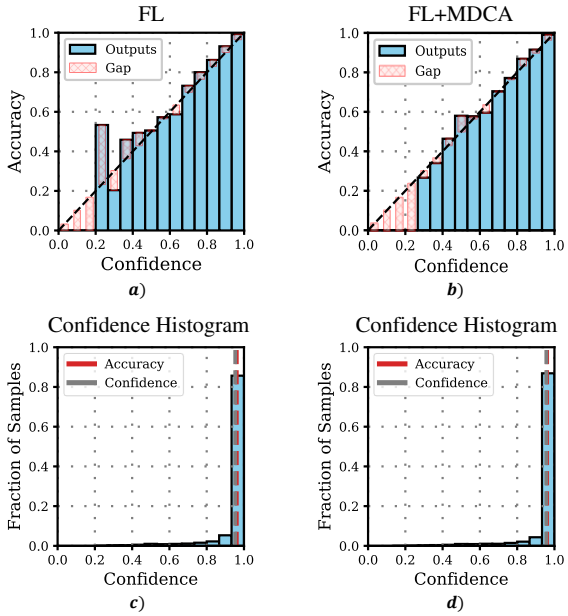

Figure 3. Reliability diagrams (a,b) and confidence histograms (c,d) of `FL` trained model compared against MDCA regularized version (`FL+MDCA`). We use ResNet-32 trained on CIFAR10 dataset for comparison.

Domain	NLL	NLL+MDCA	LS	LS+MDCA	FL	FL+MDCA
SCE (10^{-3})						
Art	6.33	5.10	7.80	5.70	8.61	6.21
Cartoon	17.95	16.53	11.95	12.07	16.22	11.91
Sketch	15.01	13.49	10.88	11.70	10.94	11.08
Average	13.10	**11.71**	10.21	**9.82**	12.06	**9.73**
ECE (%)						
Art	9.42	4.99	8.70	11.80	7.34	6.29
Cartoon	52.99	47.96	25.21	22.73	48.96	29.81
Sketch	35.56	31.68	13.29	10.34	25.33	23.05
Average	32.66	**28.21**	15.73	**14.96**	27.21	**19.71**
Top-1 (%) Accuracy						
Art	59.08	61.67	56.35	59.57	52.83	63.23
Cartoon	21.20	21.72	22.01	24.62	17.32	27.86
Sketch	28.00	26.90	29.88	26.55	26.70	23.01
Average	36.09	**36.76**	36.08	**36.91**	32.28	**38.03**

Table 5. Ablation comparing SCE, ECE and Top-1 Accuracy values of models using `MDCA` as an auxiliary loss along with other classification losses. The numbers correspond to training a ResNet-18 model on `Photo` subset from PACS dataset, and testing on other subsets of the PACS.

Method	Classes									
	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
Cross Entropy	0.98	0.43	1.12	1.97	0.65	1.47	0.65	0.44	0.58	0.57
Focal Loss [15]	0.38	**0.23**	0.69	1.08	0.39	0.91	0.37	**0.34**	0.25	**0.24**
LS [17]	1.64	1.89	1.26	1.01	1.64	1.25	1.66	1.62	1.76	1.77
Brier Score [1]	0.71	0.25	0.91	1.56	0.61	1.26	0.4	0.34	0.41	0.37
MMCE [10]	1.88	1.29	1.57	2.43	1.83	1.62	1.57	1.51	1.09	1.75
DCA [13]	0.80	0.43	1.18	1.71	0.93	1.44	0.52	0.55	0.51	0.6
FLSD [16]	0.99	1.12	0.81	1.11	0.81	1.44	0.81	0.85	0.70	1.14
Ours (FL+MDCA)	**0.36**	0.37	**0.36**	**0.60**	**0.35**	**0.59**	**0.31**	0.41	**0.25**	0.42

Table 6. Class-j-ECE (%) values on all ten classes for a ResNet-32 model trained on the CIFAR10 dataset comparing different learnable calibration methods including ours highlighted in Cyan.

References

[1] Glenn W Brier et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950. i, iv, v

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611, 2018. ii

[3] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. ii

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. ii

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. ii

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. ii

[7] Daniel Kermany, Kang Zhang, Michael Goldbaum, et al. Labeled optical coherence tomography (oct) and chest x-ray images for classification. *Mendeley data*, 2(2), 2018. ii

[8] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. i

[9] Meelis Kull, Miquel Perello-Nieto, Markus Kängsepp, Hao Song, Peter Flach, et al. Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with dirichlet calibration. *arXiv preprint arXiv:1910.12656*, 2019. iii

[10] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2805–2814. PMLR, 10–15 Jul 2018. i, iii, iv, v

[11] Ken Lang. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier, 1995. ii

[12] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. ii, iv

[13] Gongbo Liang, Yu Zhang, Xiaoqin Wang, and Nathan Jacobs. Improved trainable calibration method for neural networks on medical imaging classification. *CoRR*, abs/2009.04057, 2020. i, ii, iii, iv, v

[14] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. iii

[15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. i, iii, iv, v

[16] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip H. S. Torr, and Puneet K. Dokania. Calibrating deep neural networks using focal loss, 2020. i, ii, iv, v

[17] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? *arXiv preprint arXiv:1906.02629*, 2019. i, iii, iv, v

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

[19] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. iii

[20] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015. iv

[21] Christian Tomani, Sebastian Gruber, Muhammed Ebrar Erdem, Daniel Cremers, and Florian Buettner. Post-hoc uncertainty calibration for domain drift scenarios. *CoRR*, abs/2012.10988, 2020. ii