Beyond In-Domain Scenarios: Robust Density-Aware Calibration

Christian Tomani [*] [1] [2] **Futa Waseda** [*] [3] **Yuesong Shen** [1] [2] **Daniel Cremers** [1] [2]

Abstract

Calibrating deep learning models to yield uncertainty-aware predictions is crucial as deep neural networks get increasingly deployed in safety-critical applications. While existing post-hoc calibration methods achieve impressive results on in-domain test datasets, they are limited by their inability to yield reliable uncertainty estimates in domain-shift and out-of-domain (OOD) scenarios. We aim to bridge this gap by proposing **DAC**, an accuracy-preserving as well as **D**ensity-**A**ware **C**alibration method based on k-nearest-neighbors (KNN). In contrast to existing post-hoc methods, we utilize hidden layers of classifiers as a source for uncertainty-related information and study their importance. We show that DAC is a generic method that can readily be combined with state-of-the-art post-hoc methods. DAC boosts the robustness of calibration performance in domain-shift and OOD, while maintaining excellent in-domain predictive uncertainty estimates. We demonstrate that DAC leads to consistently better calibration across a large number of model architectures, datasets, and metrics. Additionally, we show that DAC improves calibration substantially on recent large-scale neural networks pre-trained on vast amounts of data.

1. Introduction

Deep learning models have become state-of-the-art (SOTA) in several different fields. Especially in safety-critical applications such as medical diagnosis and autonomous driving with changing environments over time, reliable model estimates for predictive uncertainty are crucial. Thus, models are required to be accurate as well as calibrated, meaning that their predictive uncertainty (or confidence) matches the expected accuracy. Due to the fact that many deep neural networks are generally uncalibrated (Guo et al., 2017), post-hoc calibration of already trained neural networks has received increasing attention in the last few years.

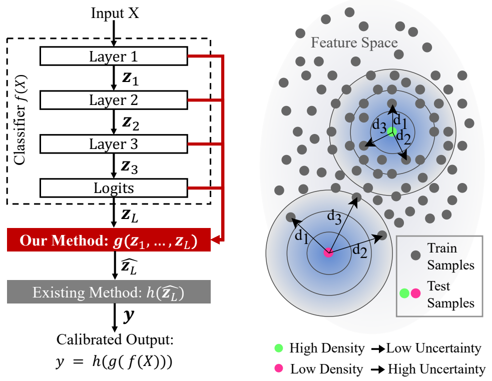


Figure 1. **Left:** Our density-aware calibration method (DAC) g can be combined with existing post-hoc methods h leading to robust and reliable uncertainty estimates. To this end, DAC leverages information from feature vectors $z_1 \ldots z_L$ across the entire classifier f. **Right:** DAC is based on KNN, where predictive uncertainty is expected to be high for test samples lying in low-density regions of the empirical training distribution and vice versa.

In order to tackle the miscalibration of neural networks, researchers have come up with a plethora of post-hoc calibration methods (Guo et al., 2017; Zhang et al., 2020; Rahimi et al., 2020b; Milios et al., 2018; Tomani et al., 2022; Gupta et al., 2021). These current approaches are particularly designed for in-domain calibration, where test samples are drawn from the same distribution as the network was trained on. Although these approaches perform almost perfect in-domain, recent works (Tomani et al., 2021) have shown that they lack substantially in providing reliable confidence scores in domain-shift and out-of-domain (OOD) scenarios (up to 1 order of magnitude worse than in-domain), which is unacceptable, particularly in safety-critical real-world scenarios where calibration of neural networks matters in order to prevent unforeseeable failures. To date, the only post-hoc methods that have been introduced to mitigate this shortcoming in domain-shift and OOD settings, use artificially created data or data from different sources in order to estimate the potential test distribution (Tomani et al., 2021;

[*] Equal contribution [1] Technical University of Munich [2] Munich Center for Machine Learning [3] The University of Tokyo. Correspondence to: Christian Tomani <christian.tomani@tum.de>, Futa Waseda <futa-waseda@g.ecc.u-tokyo.ac.jp>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

Yu et al., 2022; Wald et al., 2021; Yu et al., 2022; Gong et al., 2021). However, these methods are not generic in that they require domain knowledge about the dataset and utilize multiple domains for calibration. Additionally, they might only work well for a narrow subset of anticipated distributional shifts because they rely heavily on strong assumptions towards the potential test distribution. Furthermore, they can hurt in-domain calibration performance.

To mitigate the issue of miscalibration in scenarios where test samples are not necessarily drawn from dense regions of the empirical training distribution or are even OOD, we introduce a density-aware method that extends the field of post-hoc calibration beyond in-domain calibration. Contrary to the aforementioned existing works which focus on particularly crafted training data, our method DAC does not depend on additional data and does not rely on any assumptions about potentially shifted or out-of-domain test distributions, and is even domain agnostic. The proposed method can, therefore, simply be added to an existing post-hoc calibration pipeline, because it relies on the exact same training paradigm with a held-out in-domain validation set as current post-hoc methods do.

Previous works on calibration have focused primarily on post-hoc methods that solely take softmax outputs or logits into account (Guo et al., 2017; Zhang et al., 2020; Rahimi et al., 2020b; Milios et al., 2018; Tomani et al., 2022; Gupta et al., 2021). However, we argue that prior layers in neural networks contain valuable information for recalibration too. Moreover, we report, which layers our method identified as particularly relevant for providing well-calibrated predictive uncertainty estimates.

Recently developed large-scale neural networks that benefit from pre-training on vast amounts of data (Kolesnikov et al., 2020; Mahajan et al., 2018), have mostly been overlooked when benchmarking post-hoc calibration methods. One explanation for that could be because, e.g., vision transformers (ViTs) (Dosovitskiy et al., 2020a) are well calibrated out of the box (Minderer et al., 2021). Nevertheless, we show that also these models can profit from post-hoc methods and in particular from DAC through more robust uncertainty estimates.

1.1. Contribution

- We propose DAC, an accuracy-preserving and density-aware calibration method that can be combined with existing post-hoc methods to boost domain-shift and out-of-domain performance while maintaining in-domain calibration.[1]
- We discover that the common practice of using solely the final logits for post-hoc calibration is sub-optimal and that aggregating intermediate outputs yields improved results.
- We study recent large-scale models, such as transformers, pre-trained on vast amounts of data and encounter that also for these models our proposed method yields substantial calibration gains.

2. Related Work

Calibration methods can be divided into post-hoc calibration methods and methods that adapt the training procedure of the classifier itself. The latter one includes methods such as self-supervised learning (Hendrycks et al., 2019), Bayesian neural networks (Gal & Ghahramani, 2016; Wen et al., 2018), Deep Ensembles (Lakshminarayanan et al., 2017), label smoothing (Müller et al., 2019), methods based on synthesized feature statistics (Wang et al., 2020) or mixup techniques (Thulasidasan et al., 2019; Zhang et al., 2017) as well as other intrinsically calibrated approaches (Sensoy et al., 2018; Tomani & Buettner, 2021; Ashukha et al., 2020). Similar to post-hoc methods, Ovadia et al. (2019) have found that intrinsic methods suffer as well from miscalibration in domain-shift scenarios.

Post-hoc calibration methods, on the other hand, can be applied on top of already trained classifiers and do not require any retraining of the underlying neural network. Wang et al. (2021) argue in favour of a unified framework comprising of main training and post-hoc calibration. Rahimi et al. (2020a) provide a theoretical basis for post-hoc calibration schemes in that learning calibration functions post-hoc using a proper loss function leads to calibrated outputs.

These post-hoc calibration methods include non-parametric approaches such as histogram binning (Zadrozny & Elkan, 2001), where uncalibrated confidence scores are partitioned into bins and are assigned a respective calibrated score via optimizing a bin-wise squared loss on a validation set. Isotonic regression (Zadrozny & Elkan, 2002), which is an extension to histogram binning, fits a piecewise constant function to intervals of uncalibrated confidence scores and Bayesian Binning into Quantiles (BBQ) (Naeini et al., 2015) is different from isotonic regression in that it considers multiple binning models and their combination. In addition, Zhang et al. (2020) introduce an accuracy-preserving version of isotonic regression beyond binary tasks, which they call multi-class isotonic regression (IRM). Moreover, Wenger et al. (2020) and Milios et al. (2018) propose Gaussian processes-based calibration methods.

Approaches for training a mapping function include Platt scaling (Platt, 1999), matrix as well as vector scaling and temperature scaling (Guo et al., 2017). Temperature scaling (TS) transforms logits by a single scalar parameter in

[1] Source code available at: `https://github.com/futakw/DensityAwareCalibration`

an accuracy-preserving manner, since re-scaling does not affect the ranking of the logits. Moreover, Ensemble Temperature scaling (ETS) (Zhang et al., 2020) extends temperature scaling by two additional calibration maps with a fixed temperature of 1 and ∞, respectively. More recent and advanced approaches include Dirichlet-based scaling (Kull et al., 2019) and Parameterized Temperature scaling (Tomani et al., 2022), where a temperature is calculated sample-wise via a neural network architecture. Rahimi et al. (2020b) designed a post-hoc neural network architecture for transforming classifier logits that represent a class of intra-order-preserving functions, and Gupta et al. (2021) introduce a method for obtaining a calibration function by approximating the empirical cumulative distribution of output probabilities with the help of splines.

These post-hoc calibration methods are trained on a hold-out calibration set. Although there has been a surge of research on these approaches in recent years, Tomani et al. (Tomani et al., 2021) have discovered that post-hoc calibration methods yield highly over-confident predictions under domain-shift and are, therefore, not well suited for OOD scenarios. They introduce a strategy where samples are perturbed in the calibration set before performing the post-hoc calibration step. However, such an approach makes a strong distributional assumption on potential domain shifts during testing by perturbing training samples in a particular way, which may not necessarily hold in each case. To date, post-hoc calibration methods that are themselves capable of distinguishing in-domain samples from gradually shifted or out-of-domain samples without any distributional assumptions have not yet been addressed.

3. Method

3.1. Definitions

We study the multi-class classification problem, where $X \in \mathbb{R}^D$ denotes a D-dimensional random input variable and $Y \in \{1, 2, \ldots, C\}$ denotes the label with C classes with a ground truth joint distribution $\pi(X, Y) = \pi(Y|X)\pi(X)$. The dataset $\mathbb{D}$ contains N i.i.d. samples $\mathbb{D} = \{(X_n, Y_n)\}_{n=1}^{N}$ drawn from $\pi(X, Y)$.

Let the output of a trained neural network classifier f be $f(X) = (y, \mathbf{z}_L)$, where y denotes the predicted class and $\mathbf{z}_L$ the associated logits vector. The softmax function σ_{SM} as $p = \max_c \sigma_{SM}(\mathbf{z}_L)^{(c)}$ is then needed to transform $\mathbf{z}_L$ into a confidence score or predictive uncertainty p w.r.t y. In this paper, we propose an approach to improve the quality of the predictive uncertainty p by recalibrating the logits $\mathbf{z}_L$ from $f(X)$ via a combination of two calibration methods:

$$p = h(g(f(X))) \tag{1}$$

where g denotes our density-aware calibration method DAC rescaling logits for boosting domain-shift and OOD calibration performance and h denotes an existing state-of-the-art in-domain post-hoc calibration method (Fig. 1).

Following Guo et al. (2017), perfect calibration is defined such that confidence and accuracy match for all confidence levels:

$$\mathbb{P}(Y = y|P = p) = P, \ \ \forall P \in [0, 1] \tag{2}$$

Consequently, miscalibration is defined as the difference in expectation between accuracy and confidence.

$$\mathop{\mathbb{E}}_{P}\big[\big|\,\mathbb{P}(Y = y|P = p) - P\big|\big] \tag{3}$$

3.2. Measuring Calibration

The expected calibration error (ECE) (Naeini et al., 2015) is frequently used for quantifying miscalibration. ECE is a scalar summary measure estimating miscalibration by approximating equation (3) as follows. In the first step, confidence scores $\hat{\mathcal{P}}$ of all samples are partitioned into M equally sized bins of size $1/M$, and secondly, for each bin B_m the respective mean confidence and the accuracy is computed based on the ground truth class y. Finally, the ECE is estimated by calculating the mean difference between confidence and accuracy over all bins:

$$\mathrm{ECE}^d = \sum_{m=1}^{M} \frac{|B_m|}{N} \left\|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right\|_d \tag{4}$$

with d usually set to 1 (L1-norm).

3.3. Density-Aware Calibration (DAC)

Our main idea for our proposed calibration method g stems from the fact that test samples lying in high-density regions of the empirical training distribution can generally be predicted with higher confidence than samples lying in low-density regions. For the latter case, the network has seen very few, if any, training samples in the neighborhood of the respective test sample in feature space, and is thus not able to provide reliable predictions for those samples. Leveraging this information about density through a proxy can result in better calibration.

In order to estimate such a proxy for each sample, we propose to utilize non-parametric density estimation using k-nearest-neighbor (KNN) based on feature embeddings extracted from the classifier. KNN has successfully been applied in out-of-distribution detection (Sun et al., 2022). In contrast to Sun et al. (2022), who only take the penultimate layer into account, we argue that prior layers yield important information too, and therefore, incorporate them in our method as follows. We call our method Density-Aware Calibration (DAC).

Temperature scaling (Guo et al., 2017) is a frequently used calibration method, where a single scalar parameter T is used to re-scale the logits of an already trained classifier in order to obtain calibrated probability estimates $\hat{Q}$ for logits $\mathbf{z}_L$ using the softmax function σ_{SM} as

$$\hat{Q} = \sigma_{SM}(\mathbf{z}_L/T) \tag{5}$$

Similar to temperature scaling, our method is also accuracy preserving, in that we use one parameter $S(\mathbf{x}, w)$ for rescaling the logits of the classifier:

$$\hat{Q}(\mathbf{x}, w) \quad = \quad \sigma_{SM}(\mathbf{z}_L/S(\mathbf{x}, w)) \tag{6}$$

In contrast to temperature scaling, in our case, $S(\mathbf{x}, w)$ is sample-dependent with respect to $\mathbf{x}$ and is calculated via a linear combination of density estimates s_l as follows:

$$S(\mathbf{x}, w) = \sum_{l=1}^{L} w_l s_l + w_0 \tag{7}$$

with $w_1 ... w_L$ being the weights for every layer in L and w_0 being a bias term. Note that only positive weights are valid because negative weights would assign high confidence to outliers. Thus, we constrain the weights to be positive to tackle overfitting.

For each feature layer l, we compute the density estimate s_l in the neighborhood of the empirical training distribution of the respective test sample $\mathbf{x}$ with the k-th nearest neighbor distance: First, we derive the test feature vector $\mathbf{z}_l$ from the trained classifier f given the test input sample $\mathbf{x}$ and average over spatial dimensions as well as normalize it. We then use the normalized training feature vectors $Z_{N_{Tr},l} = (\mathbf{z}_{1,l}, \mathbf{z}_{2,l}, ..., \mathbf{z}_{N_{Tr},l})$, which we gathered from the training dataset $X_{N_{Tr}} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{N_{Tr}})$ to calculate the euclidean distance between $\mathbf{z}_l$ and each element in $Z_{N_{Tr},l}$ for each sample i in the training set N_{Tr} as follows:

$$d_{i,l} = \|\mathbf{z}_{i,l} - \mathbf{z}_l\| \tag{8}$$

The resulting sequence $D_{N_{Tr},L} = (d_{1,l}, d_{2,l}, ..., d_{N_{Tr},l})$ is reordered. Finally, s_l is given by the k-th smallest element (=k-th nearest neighbor) in the sequence: $s_l = d_{(k)}$, with (k) indicating the index in the reordered sequence $D_{N_{Tr},L}$. For determining k, we follow Sun et al. (2022), who did a thorough analysis and concluded that a proper choice for k is 50 for CIFAR10, 200 for CIFAR100 for all training samples and 10 for ImageNet for 1% training samples.

We fit our method for a trained neural network $f(X) = (y, \mathbf{z}_L)$ by optimizing a squared error loss L_w w.r.t. w.

$$L_w \quad = \quad \sum_{c=1}^{C} (I_c - \sigma_{SM}(\mathbf{z}_L/S(\mathbf{x}, w))^{(c)})^2 \tag{9}$$

where I_c indicates a binary variable, which is 1 if the respective sample has true class c, and 0 otherwise. We accumulate L_w over all samples in the validation set.

The rescaled logits $\hat{\mathbf{z}}_L(\mathbf{x}, w) = \mathbf{z}_L/S(\mathbf{x}, w)$ and consequently the recalibrated probability estimates $\hat{Q}(\mathbf{x}, \mathbf{w})$ can directly be fed to another post-hoc method. Thus, DAC can be applied prior to other existing in-domain post-hoc calibration methods for robustly calibrating models in domain-shift and OOD scenarios (Fig. 1).

DAC uses KNN (a non-parametric method) to compute a density proxy per layer and combines these proxies linearly across layers, whereas other methods like Parameterized Temperature scaling (Tomani et al., 2022) and models that utilize intra-order-preserving functions for calibration (Rahimi et al., 2020b) are parametric methods using a neural network. Moreover, DAC uses intermediate features from hidden layers and is particularly designed with domain shift and OOD calibration behavior in mind.

Our method has the following advantages:

- **Density aware:** Due to distance-based density estimation across feature layers, our method is capable of inferring how close or how far a test sample is in feature space with respect to the training distribution and can adjust the predictive estimates of the classifier accordingly.
- **Domain agnostic:** Since we use KNN, a non-parametric method for density estimation, no distributional assumptions are imposed on the feature space, and our method is therefore applicable to any type of in-domain, domain-shift, or OOD scenario.
- **Backbone agnostic:** Adapts easily to different underlying classifier architectures (e.g., CNNs, ResNets, and more recent models like transformers) because during training, DAC automatically figures out the informative feature layers regarding uncertainty calibration.

4. Experimental Setup

Models and Datasets In our study, we quantify the performance of our proposed model for various model architectures and different datasets. We consider 3 different datasets to evaluate our models on CIFAR10/100 (Krizhevsky et al., 2009), and ImageNet-1k (Deng et al., 2009). In particular, for measuring performance on CIFAR10 and CIFAR100, we train ResNet18 (He et al., 2016a), VGG16 (Simonyan & Zisserman, 2014) and DenseNet121 (Huang et al., 2017), and for ImageNet, we use 3 pre-trained models, namely ResNet152 (He et al., 2016a), DenseNet169 (Huang et al., 2017), and Xception(Chollet, 2017). We further investigate post-hoc calibration methods applied to new state-of-the-art architectures as well as modern training schemes. To this

Table 1. Mean expected calibration error across all test domain-shift scenarios. For each model, the macro-averaged ECE ($\times 10^2$) (with equal-width binning and 15 bins) is computed across all corruptions from severity=0 (in-domain) until severity=5 (heavily corrupted). Post-hoc calibration methods paired with our method are consistently better calibrated than simple post-hoc methods. (lower ECE is better)

	UNCAL	BASELINE CALIBRATION METHODS					COMBINATION WITH DAC (OURS)				
	-	TS	ETS	IRM	DIA	SPL	TS	ETS	IRM	DIA	SPL
C10 RESNET18	19.27	4.96	4.96	5.93	6.39	6.27	4.49	**4.39**	4.90	4.61	4.74
C10 VGG16	19.05	6.25	6.30	7.45	9.82	7.57	**5.66**	**5.66**	6.30	6.33	5.69
C10 DENSENET121	19.26	5.21	5.21	6.66	7.81	6.60	4.59	4.59	5.69	6.60	**4.42**
C100 RESNET18	16.44	11.37	10.56	12.26	9.24	10.40	10.66	8.96	10.04	**8.47**	9.74
C100 VGG16	34.41	11.54	11.54	13.24	14.62	10.76	6.49	**6.48**	8.11	10.26	7.45
C100 DENSENET121	23.83	8.80	8.76	12.07	11.02	9.73	8.77	**8.40**	10.05	15.01	9.15
IMG RESNET152	10.50	4.47	4.01	5.20	7.17	5.56	3.48	**3.34**	3.50	3.64	3.63
IMG DENSENET169	13.28	6.59	6.34	7.37	8.44	7.12	4.81	**3.87**	4.53	6.31	4.60
IMG XCEPTION	30.49	8.81	8.40	12.93	9.83	10.80	8.79	**7.99**	8.38	8.99	8.49
IMG BIT-M	11.71	7.17	6.56	6.93	7.45	6.62	4.40	3.98	4.21	5.51	**3.76**
IMG RESNEXT-WSL	15.44	8.03	8.03	8.04	8.32	6.16	7.32	5.63	5.75	6.32	**3.90**
IMG VIT-B	3.78	4.23	3.72	4.24	5.85	3.93	3.80	**3.34**	3.99	5.52	3.56

end, we include the following models in our study, which are all finetuned on ImageNet-1k:

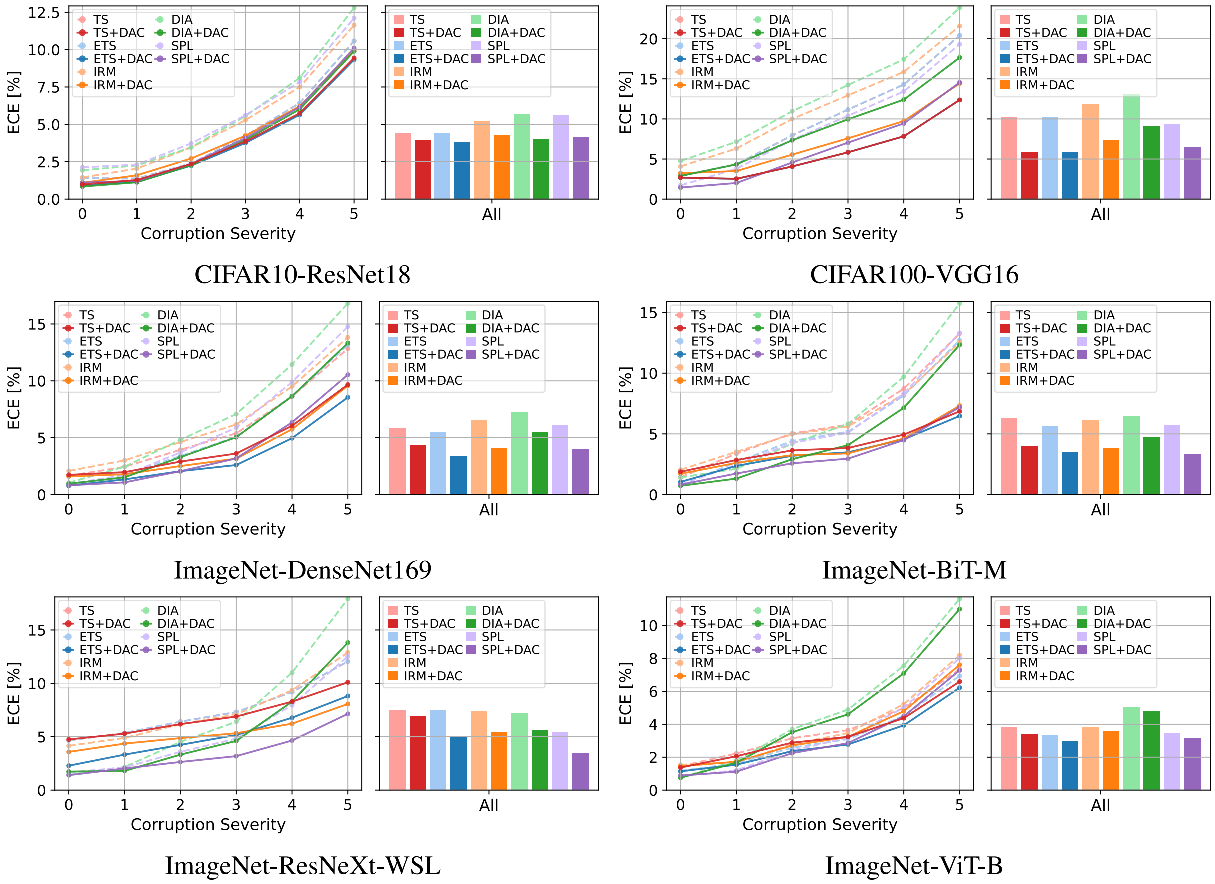
- **BiT-M** (Kolesnikov et al., 2020): Is a ResNet-based architecture (ResNetV2) (He et al., 2016b) pre-trained on ImageNet-21k.
- **ResNeXt-WSL** (Mahajan et al., 2018): Is a ResNeXt-based architecture (ResNeXt101 32x8d) (Xie et al., 2017), which is weakly supervised pre-trained with billions of hashtags of social media images.
- **ViT-B** (Dosovitskiy et al., 2020b): Is a transformer-based architecture pre-trained on ImageNet-21k.

We quantify calibration performance for in-domain, domain-shift, and OOD scenarios. In order to ensure a gradual domain shift in our evaluation pipeline, we use ImageNet-C as well as CIFAR-C (Hendrycks & Dietterich, 2019), which were specifically developed to produce domain shift and were incorporated in many related studies since. Both datasets have 18 distinct corruption types, each having 5 different levels of severity, mimicking a scenario where the input data to a classifier gradually shifts away from the training distribution. Additionally, we test our models on a real-world OOD dataset, namely ObjectNet-OOD. ObjectNet (Barbu et al., 2019) is a dataset consisting of 50,000 test images with a total of 313 classes, of which 200 classes are out-of-domain with respect to ImageNet. Hence, we make use of these 200 classes for our OOD analysis.

Post-hoc Calibration Methods We consider the currently best performing post-hoc calibration methods for benchmarking as well as for combining them with DAC: Temperature scaling (TS) (Guo et al., 2017), Ensemble Temperature scaling (ETS) (Zhang et al., 2020), accuracy preserving version of Isotonic Regression (IRM) (Zhang et al., 2020), Intra-order preserving calibration (DIA) (Rahimi et al., 2020b), Calibration using Splines (SPL) (Gupta et al., 2021). Additionally, we show results for Isotonic Regression (IR) (Zadrozny & Elkan, 2002), Parameterized Temperature scaling (PTS) (Tomani et al., 2022) and Dirichlet calibration (DIR) (Kull et al., 2019) in Appendix D.

Our proposed method DAC does not solely rely on logits for calibration as other post-hoc calibration approaches do; it rather takes various layers at certain positions of the network into account. Even though DAC could use every layer in a classifier due to its weighting scheme, we opt for a much simpler and faster version. That is, we follow a structured approach for choosing layers, e.g., after each ResNet or transformer block. A detailed description of which layers we use can be found in Appendix C.1. In the results, we show that our selective approach produces similar results compared to taking all layers into account.

Measuring Calibration Our evaluation is based on various calibration measures. Throughout the paper, we provide results for ECE and Brier scores using equal-width binning with 15 bins. Although ECE is the most commonly used metric for evaluating and comparing post-hoc calibration methods, it bears several limitations. That is why, in Appendix E, we show that our results hold for different kinds of calibration measures, including ECE based on kernel density estimation (ECE-KDE) (Zhang et al., 2020), ECE using equal-mass binning and class-wise ECE(Kull et al., 2019) as well as we demonstrate consistency with likelihood.



CIFAR10-ResNet18

CIFAR100-VGG16

ImageNet-DenseNet169

ImageNet-BiT-M

ImageNet-ResNeXt-WSL

ImageNet-ViT-B

Figure 2. Expected calibration error ($\times 10^2$) of post-hoc methods with and without our method DAC for different model and dataset combinations. **Line plots**: Macro-averaged ECE across all corruption types shown for each corruption severity, from in-domain to severity=5 (heavily corrupted). **Bar plots**: Macro-averaged ECE across all corruption types as well as across all severities. Our model captures domain-shift scenarios reliably and thus increases calibration (=decreases ECE) across the whole spectrum of corruptions.

5. Results

First, we show that combining our method DAC with existing post-hoc calibration methods increases calibration performance across the entire spectrum, from in-domain to heavily corrupted data distributions, for different datasets and various model architectures, including transformers. Secondly, we show the calibration performance of our method on purely OOD scenarios. Lastly, we conduct additional experiments, such as a layer importance analysis and a data efficiency analysis.

5.1. DAC Boosts Calibration Performance Beyond In-Domain Scenarios

We begin by systematically assessing whether the performance of state-of-the-art post-hoc calibration methods can be improved when extended by our proposed DAC method. In particular, we are interested in scenarios under domain shift. To this end, we show calibration performance on CIFAR-C and ImageNet-C for severity levels from 1 to 5 and additionally provide results for in-domain scenarios (severity=0). Fig. 2 illustrates a comparison between standalone post-hoc methods and combined methods with DAC for various classifiers. The line charts reveal that DAC consistently improves the calibration performance of post-hoc methods in domain-shift cases. Tab. 2 underpins this performance increase even further by revealing a substantial decrease in the absolute ECE for heavily corrupted data (severity=5) scenarios when DAC is used. When optimizing for domain-shift performance, a sharp decline in in-domain performance is generally observed for other existing methods (Tomani et al., 2021). This is, however not the case for our method, where we even observe slight improvements of in-domain ECE for many classifier and post-hoc method configurations, and for the few cases where in-domain ECE marginally increases (in the order of 10^{-4} to 10^{-3}), the ECE in heavily corrupted scenarios decreases in the order of far more than a magnitude in comparison. Finally, in order to measure the overall improvement of DAC across the entire spectrum of corruption from severity 0 to 5, we

Table 2. Difference in expected calibration error ($\times 10^2$) of post-hoc calibration methods with and without our method DAC. InD: In-domain, $SEV.5$: Heavily corrupted (severity of 5) and ALL: Macro-averaged ECE across all corruptions from severity=0 (in-domain) until severity=5 (heavily corrupted). For ALL, we additionally report the ratio to indicate overall performance gain. Post-hoc calibration methods combined with DAC consistently improve calibration in cases of heavy corruption as well as overall calibration (relative improvement of around 5-40%), while preserving in-domain performance (negative deltas are better).

	IND	SEV. 5	ALL	
TS	-0.42	-1.14	-0.46	10%
ETS	-0.49	-1.25	-0.56	13%
IRM	-0.36	-1.67	-0.92	18%
DIA	-1.09	-2.91	-1.67	29%
SPL	-1.02	-2.00	-1.45	26%

A.) CIFAR10 RESNET18

	IND	SEV. 5	ALL	
TS	-0.22	-8.05	-4.29	42%
ETS	-0.22	-8.06	-4.29	42%
IRM	-0.85	-7.15	-4.46	38%
DIA	-1.87	-6.20	-3.97	30%
SPL	-0.27	-4.77	-2.83	30%

B.) CIFAR100 VGG16

	IND	SEV. 5	ALL	
TS	+0.08	-3.16	-1.49	26%
ETS	+0.03	-4.61	-2.08	38%
IRM	-0.48	-4.25	-2.47	38%
DIA	-0.09	-3.49	-1.81	25%
SPL	-0.15	-4.23	-2.15	35%

C.) IMAGENET DENSENET169

	IND	SEV. 5	ALL	
TS	+0.47	-6.40	-2.26	36%
ETS	+0.28	-6.23	-2.13	38%
IRM	-0.34	-5.20	-2.34	38%
DIA	-0.69	-3.42	-1.75	27%
SPL	-0.01	-6.11	-2.41	42%

D.) IMAGENET BIT-M

	IND	SEV. 5	ALL	
TS	+0.09	-1.95	-0.58	8%
ETS	-2.37	-3.24	-2.39	32%
IRM	-0.57	-4.82	-2.02	27%
DIA	+0.30	-4.10	-1.64	23%
SPL	-0.25	-5.39	-1.94	36%

E.) IMAGENET RESNEXT-WSL

	IND	SEV. 5	ALL	
TS	-0.08	-0.74	-0.38	10%
ETS	-0.03	-0.72	-0.32	10%
IRM	-0.08	-0.59	-0.23	6%
DIA	-0.00	-0.62	-0.28	6%
SPL	+0.02	-0.74	-0.30	9%

F.) IMAGENET VIT-B

calculate the macro-averaged ECE across all corruption types and levels of severity for each method. We discover a consistent improvement for all post-hoc methods combined with DAC as opposed to stand-alone methods visualized in the bar charts in Fig. 2 and in Tab. 2. Moreover, Tab. 1 further reveals that DAC consistently boosts calibration performance across diverse architectures and datasets. In Tab. 3 we demonstrate that improved calibration performance is also consistent with better Brier scores.

Since research regarding state-of-the-art post-hoc methods applied to recent large-scale neural networks is still lacking, we want to attempt to bridge this gap and show results for modern ResNet as well as transformer architectures trained on large corpora of data. We make the same observation as Minderer et al. (Minderer et al., 2021) that, in fact, even modern ResNet architectures are not well calibrated in domain-shift settings despite being pre-trained on huge amounts of data, yet transformer architectures perform particularly well. In our experiments, we observe that modern ResNet architectures (BiT-M and ResNeXt) can indeed be further calibrated with post-hoc methods. Especially, SPL+DAC performs best and reduces the ECE by around 37% for ResNeXt-WSL and 42% for BiT-M compared to the best-performing standard post-hoc method (Tab. 1). ViT-B, on the other hand, can profit from existing post-hoc methods too, at least in-domain. For domain-shift scenarios, ViT outperforms all standard post-hoc methods, except when combined with DAC, in that case, ETS+DAC outperforms existing methods by 12%.

5.2. Calibration in OOD Scenarios

To complement the previous distributional shift experiments, we conduct additional experiments for the out-of-domain case, incorporating data samples with completely different classes w.r.t. the training data. Ideally, a well-calibrated uncertainty-aware model would produce high-confidence predictions for in-domain data and low-confidence ones for the OOD case, allowing for the detection of OOD data samples. Based on this idea, several metrics have been proposed in the OOD literature (Hendrycks & Gimpel, 2017; Liang et al., 2018; Lee et al., 2018) to quantify the model performance in OOD scenarios, including FPR at 95% TPR, detection error, AUROC, AUPR-In/AUPR-Out, which we employ for our experiments.

To examine DAC in the OOD scenario, we set up the OOD dataset with in-domain data from ImageNet-1k and OOD data from ObjectNet-OOD (c.f. Section 4 for dataset descriptions). Top-class confidence predictions are produced by models trained and calibrated on ImageNet-1k with various calibration methods with and without the proposed DAC method. The OOD metrics are computed from the confidence predictions. We summarize the results for the DenseNet169 backbone in Tab. 4. Additional results for other backbones can be found in Appendix H.

In general, we see that DAC yields more robust calibration in the OOD scenario, as demonstrated by its consistent improvement of OOD results.

Table 3. Mean **Brier-score** computed across all corruptions from severity=0 (in-domain) until severity=5 (heavily corrupted). Note that since SPL only calibrates the highest predicted confidence, it is not directly possible to evaluate Brier scores.

	UNCAL	BASELINE CALIBRATION METHODS				COMBINATION WITH DAC (OURS)			
	-	TS	ETS	IRM	DIA	TS	ETS	IRM	DIA
C10 RESNET18	0.4547	0.3865	0.3865	0.3913	0.3906	0.3842	**0.3841**	0.3852	0.3842
C10 VGG16	0.4294	0.3562	0.3564	0.3633	0.3717	**0.3547**	**0.3547**	0.3559	0.3565
C10 DENSENET121	0.4593	0.3940	0.3940	0.3978	0.3986	**0.3896**	**0.3896**	0.3900	0.3926
C100 RESNET18	0.6902	0.6655	0.6638	0.6696	0.6568	0.6655	0.6562	0.6584	**0.6525**
C100 VGG16	0.8205	0.6784	0.6784	0.6837	0.6873	**0.6554**	**0.6554**	0.6578	0.6652
C100 DENSENET121	0.6963	0.6261	0.6265	0.6370	0.6312	0.6261	**0.6254**	0.6275	0.6460
IMG RESNET152	0.6510	0.6360	0.6354	0.6432	0.6432	**0.6331**	**0.6331**	0.6332	**0.6331**
IMG DENSENET169	0.6707	0.6505	0.6507	0.6582	0.6562	0.6457	**0.6443**	0.6460	0.6494
IMG XCEPTION	0.8744	0.7539	0.7534	0.7812	0.7547	0.7539	**0.7519**	0.7533	0.7521
IMG BIT-M	0.6286	0.6173	0.6164	0.6184	0.6134	0.6067	0.6064	**0.6057**	0.6064
IMG RESNEXT-WSL	0.5393	0.5117	0.5117	0.5143	0.5144	0.5090	0.5074	**0.5023**	0.5068
IMG VIT-B	0.5809	0.5816	0.5812	0.5815	0.5869	0.5809	**0.5806**	0.5813	0.5862

Table 4. OOD performance with DenseNet169 trained on ImageNet-1k and using ImageNet-1k/ObjectNet-OOD as in-domain/OOD test sets, respectively. We observe that DAC consistently improves all the OOD metrics for all baseline methods.

	FPR @95% ↓	DET. ERR ↓	AU-ROC ↑	AUPR-IN ↑	AUPR-OUT ↑
TS	21.47	22.86	84.92	81.14	86.87
+DAC	**20.73**	**22.30**	**85.50**	**81.86**	**87.46**
ETS	21.92	23.15	84.72	79.82	86.72
+DAC	**20.90**	**22.29**	**85.51**	**82.16**	**87.47**
IRM	21.51	22.31	83.62	81.22	86.10
+DAC	**4.87**	**20.72**	**85.81**	**82.73**	**87.90**
DIA	22.09	24.26	83.67	79.86	85.77
+DAC	**21.36**	**23.64**	**84.35**	**80.63**	**86.47**
SPL	21.66	24.38	83.65	79.95	85.78
+DAC	**20.89**	**22.30**	**85.54**	**82.05**	**87.56**

5.3. Layer Importance for Calibration

Next, we want to investigate which layers of each classifier carry valuable information for DAC to yield calibrated predictions. For each layer, DAC learns a weight based on the importance of the respective layer (equation (7)). In Fig. 3 we demonstrate that DAC focuses on a few important layers, yet the logits layer is never one of them. This is particularly interesting because current state-of-the-art post-hoc calibration methods only focus on the logits vector for recalibration without even considering hidden layers of the classifier. Hence, we can conclude that one reason for DAC's performance improvement can be attributed to its ability to take information from other layers into account apart from the logits layer.

Even though the layers DAC has access to are well distributed throughout the architecture of the classifier, we want to investigate whether DAC can capture all the necessary information present in all the layers of the classifier. To this end, we compare our simple and fast DAC method, which uses a subset of the layers, to a holistic DAC, which utilizes all layers. In Fig. 4, we illustrate the weights DAC assigns to every layer, which are normalized to add up to 1. The holistic DAC is able to attend to various layers; however, we observe that this does not necessarily result in better calibration performance, which can be attributed to overfitting (see Appendix F for further insights).

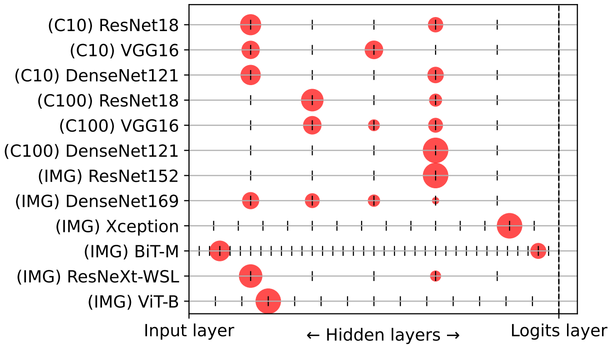


Figure 3. Importance of classifier layers found by DAC. The size of blobs indicates the magnitude of assigned weights for each layer after training of DAC from left (input) to right (logits).

5.4. Sensitivity Analysis of KNN

In this section, we evaluate the sensitivity of DAC to the hyperparameter k used for KNN operations in each layer. To this end, we study the resulting calibration performance while varying k. Figure 5 illustrates the impact of k on the calibration performance of DAC when combined with ETS and SPL, with DAC having different values of $k \in \{1, 10, 50, 100, 200\}$. For each model, the macro-averaged ECE

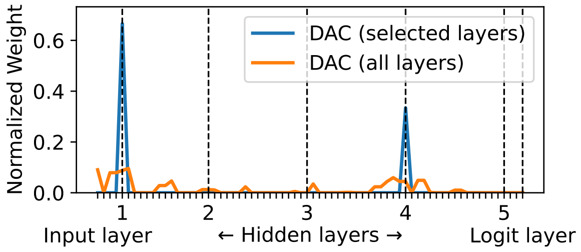


Figure 4. Comparison between our DAC with "selected layers" to a holistic DAC, which utilizes "all layers" present in ResNet18 trained on CIFAR10. DAC is able to capture the most relevant areas in layer space with important information for calibration.

($\times 10^2$) is computed across all corruptions from severity=0 (in-domain) until severity=5 (heavily corrupted). We show that DAC consistently boosts the performance of ETS and SPL regardless of the choice of k, indicating that the performance of DAC is not overly sensitive to the value of k. On the other hand, we observe that using the proper choice for k, suggested by (Sun et al., 2022), indeed results in the best calibration performance also for our method, implying that putting in additional effort for hyperparameter tuning can further enhance the performance of DAC.

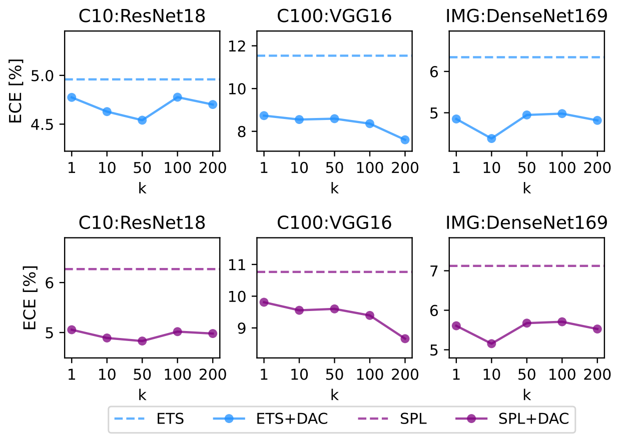


Figure 5. The sensitivity of DAC to the hyperparameter k used for KNN operations in each layer. We combined DAC with ETS (1st row) and SPL (2nd row), varying hyperparameter k.

5.5. Data Efficiency of DAC

Lastly, we investigate how sensitive our method is to various validation set sizes. We focus on the best-performing methods, namely ETS+DAC and SPL+DAC, and compare them to the respective stand-alone methods, ETS and SPL. Additionally, we incorporate TS in our study since this method is least likely to suffer from overfitting because it comprises only one parameter to train. We encounter in Fig. 6 that no matter which validation set size we use, combinations with DAC perform better than without DAC. Additionally, DAC does not overfit the data for small validation set sizes.

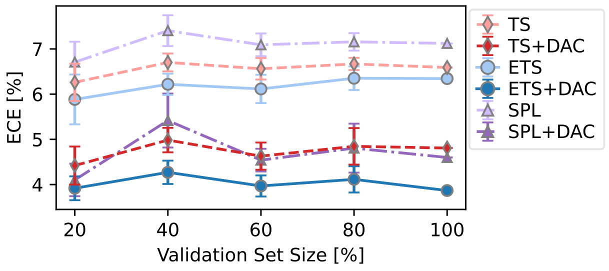


Figure 6. DAC is robust across different validation set sizes (10% - 100%) for DenseNet169 trained on ImageNet (ECE ($\times 10^2$)). We conducted five experiments with randomly sampled validation sets.

6. Conclusion

In this work, we have introduced an accuracy-preserving density-aware calibration method that can readily be applied with SOTA post-hoc methods in order to boost domain-shift and OOD calibration performance. We found that our proposed method DAC combined with existing post-hoc calibration methods yields robust predictive uncertainty estimates for any level of domain shift, from in-domain to truly OOD scenarios. In particular, ETS+DAC, as well as SPL+DAC, performed the best. We further demonstrated that hidden layers in classifiers carry valuable information for accurately predicting uncertainty estimates. Lastly, we show that even recently developed large-scale models pretrained on vast amounts of data can be calibrated effectively by DAC, opening up new research directions within the field of post-hoc calibration for entirely new applications. One of the limitations of our method can arise when applying it to highly parametric classifiers with numerous layers, as well as when determining which layers possess the most calibration-related information for DAC. The amount of calibration-related information within specific layers of a classifier seems to depend not only on the model architecture but also on the relationship between model size and dataset characteristics. We hope our findings will encourage further research into developing post-hoc methods that take into account features from the underlying neural network classifier, rather than just the output features.

References

Ashukha, A., Lyzhov, A., Molchanov, D., and Vetrov, D. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *arXiv preprint arXiv:2002.06470*, 2020.

Barbu, A., Mayo, D., Alverio, J., Luo, W., Wang, C., Gutfreund, D., Tenenbaum, J., and Katz, B. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in neural information processing systems*, 32, 2019.

Chen, Q. and Koltun, V. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 1511–1520, 2017.

Chollet, F. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020a.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020b.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Gatys, L. A., Ecker, A. S., and Bethge, M. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016.

Gong, Y., Lin, X., Yao, Y., Dietterich, T. G., Divakaran, A., and Gervasio, M. Confidence calibration for domain generalization under covariate shift. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8958–8967, 2021.

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

Gupta, K., Rahimi, A., Ajanthan, T., Mensink, T., Sminchisescu, C., and Hartley, R. Calibration of neural networks using splines. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=eQe8DEWNN2W`.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016a.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016b.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. *International Conference on Learning Representations 2019*, 2019.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017.

Hendrycks, D., Mazeika, M., Kadavath, S., and Song, D. Using self-supervised learning can improve model robustness and uncertainty. *Advances in neural information processing systems*, 32, 2019.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Johnson, J., Douze, M., and Jégou, H. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7 (3):535–547, 2019.

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., and Houlsby, N. Big transfer (bit): General visual representation learning. In *European conference on computer vision*, pp. 491–507. Springer, 2020.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kull, M., Perello Nieto, M., Kängsepp, M., Silva Filho, T., Song, H., and Flach, P. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in neural information processing systems*, 32, 2019.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018.

Liang, S., Li, Y., and Srikant, R. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. A simple baseline for bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Mahajan, D. K., Girshick, R. B., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., and van der Maaten, L. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018.

Milios, D., Camoriano, R., Michiardi, P., Rosasco, L., and Filippone, M. Dirichlet-based gaussian processes for large-scale calibrated classification. *Advances in Neural Information Processing Systems*, 31, 2018.

Minderer, M., Djolonga, J., Romijnders, R., Hubis, F., Zhai, X., Houlsby, N., Tran, D., and Lucic, M. Revisiting the calibration of modern neural networks. *Advances in Neural Information Processing Systems*, 34:15682–15694, 2021.

Müller, R., Kornblith, S., and Hinton, G. E. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

Naeini, M. P., Cooper, G., and Hauskrecht, M. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pp. 13991–14002, 2019.

Platt, J. C. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers*, pp. 61–74. MIT Press, 1999.

Rahimi, A., Gupta, K., Ajanthan, T., Mensink, T., Sminchisescu, C., and Hartley, R. Post-hoc calibration of neural networks. *arXiv preprint arXiv:2006.12807*, 2020a.

Rahimi, A., Shaban, A., Cheng, C.-A., Hartley, R., and Boots, B. Intra order-preserving functions for calibration of multi-class neural networks. *Advances in Neural Information Processing Systems*, 33:13456–13467, 2020b.

Sensoy, M., Kaplan, L., and Kandemir, M. Evidential deep learning to quantify classification uncertainty. In *Advances in Neural Information Processing Systems*, pp. 3179–3189, 2018.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Sun, Y., Ming, Y., Zhu, X., and Li, Y. Out-of-distribution detection with deep nearest neighbors. *arXiv preprint arXiv:2204.06507*, 2022.

Thulasidasan, S., Chennupati, G., Bilmes, J. A., Bhattacharya, T., and Michalak, S. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Tomani, C. and Buettner, F. Towards trustworthy predictions from deep neural networks with fast adversarial calibration. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, 2021.

Tomani, C., Gruber, S., Erdem, M. E., Cremers, D., and Buettner, F. Post-hoc uncertainty calibration for domain drift scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10124–10132, 2021.

Tomani, C., Cremers, D., and Buettner, F. Parameterized temperature scaling for boosting the expressive power in post-hoc uncertainty calibration. In *European Conference on Computer Vision*, pp. 555–569. Springer, 2022.

Wald, Y., Feder, A., Greenfeld, D., and Shalit, U. On calibration and out-of-domain generalization. *Advances in neural information processing systems*, 34:2215–2227, 2021.

Wang, D.-B., Feng, L., and Zhang, M.-L. Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. *Advances in Neural Information Processing Systems*, 34:11809–11820, 2021.

Wang, X., Long, M., Wang, J., and Jordan, M. Transferable calibration with lower bias and variance in domain adaptation. *Advances in Neural Information Processing Systems*, 33:19212–19223, 2020.

Wen, Y., Vicol, P., Ba, J., Tran, D., and Grosse, R. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *arXiv preprint arXiv:1803.04386*, 2018.

Wenger, J., Kjellström, H., and Triebel, R. Non-parametric calibration for classification. In *International Conference on Artificial Intelligence and Statistics*, pp. 178–190. PMLR, 2020.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

Yu, Y., Bates, S., Ma, Y., and Jordan, M. I. Robust calibration with multi-domain temperature scaling. *arXiv preprint arXiv:2206.02757*, 2022.

Zadrozny, B. and Elkan, C. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, pp. 609–616. Citeseer, 2001.

Zadrozny, B. and Elkan, C. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 694–699, 2002.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zhang, J., Kailkhura, B., and Han, T. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International Conference on Machine Learning (ICML)*, 2020.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

A. Datasets

We follow the standard setup for evaluating calibration performance (Guo et al., 2017). We split each dataset into train, validation, and test set. The train set is used to train classifier neural networks. The validation set is used to optimize the post-hoc calibration methods to recalibrate classifiers. Then, the calibration performances are evaluated on the test set.

In Tab. 5, we show the numbers of image-label pairs in each split we use. For CIFAR10 and CIFAR100 (Krizhevsky et al., 2009), following Guo et al. (2017), we split the original train set, which contains 50,000 image-label pairs, into 45,000 image-label pairs of train set and 5,000 image-label pairs of the validation set. For ImageNet (Deng et al., 2009)we split the original validation set, which contains 50,000 image-label pairs, into 12,500 image-label pairs of the validation set and 37,500 image-label pairs of the test set. As a common practice, we split the official validation set from ImageNet into a validation set for training our post-hoc method and test set (Minderer et al., 2021).

Table 5. The numbers of image-label pairs we used for each dataset.

DATASET	TRAIN	VAL	TEST
CIFAR10	45,000	5,000	10,000
CIFAR100	45,000	5,000	10,000
IMAGENET	1,281,167	12,500	37,500

To quantify calibration performance for domain-shift scenarios, we use ImageNet-C as well as CIFAR-C (Hendrycks & Dietterich, 2019). Both datasets have 18 distinct corruption types, each having 5 different levels of severity, mimicking a scenario where the input data to a classifier gradually shifts away from the training distribution. The 18 corruptions include 4 noise corruptions (gaussian noise, shot noise, speckle noise, and impulse noise), 4 blur corruptions (defocus blur, gaussian blur, motion blur, and zoom blur), 5 weather corruptions (snow, fog, brightness, spatter, and frost), and 5 digital corruptions (elastic transform, pixelate, JPEG compression, contrast, and saturate).

B. Classifiers

In this section we describe the implementation of the classifiers we used in our work.

- CIFAR10:
  - ResNet18 (He et al., 2016a)/VGG16 (Simonyan & Zisserman, 2014)/DenseNet121 (Huang et al., 2017): We use PyTorch's official implementation to obtain the model architectures. We trained all models for 200 epochs: 100 epochs at a learning rate of 0.01, 50 epochs at a learning rate of 0.005, 30 epochs at a learning rate of 0.001, and 20 epochs at a learning rate of 0.0001. We use a basic data augmentation technique of random cropping and horizontal flipping.
- CIFAR100:
  - ResNet18 (He et al., 2016a)/VGG16 (Simonyan & Zisserman, 2014)/DenseNet121 (Huang et al., 2017): We obtain architectures from a github repository[2] that provides PyTorch implementation of the optimal architectures for CIFAR100 dataset. We trained all models for 200 epochs, with an initial learning rate of 0.01, which decays 0.2 times at the 60, 120, and 160th epochs. We use a basic data augmentation technique of random cropping and horizontal flipping.
- ImageNet-1k: We use pre-trained models.
  - ResNet152 (He et al., 2016a)/DenseNet169 (Huang et al., 2017): We use the ImageNet-1k pre-trained models from the torchvision library.
  - Xception (Chollet, 2017): We use the ImageNet-1k pre-trained model provided by the timm library[3].
  - BiT-M (Kolesnikov et al., 2020): We use the ImageNet-21k pre-trained and ImageNet-1k fine-tuned model provided by the timm library[3]. Specifically, we use the BiT-M based on ResNetV2 101x1 architecture.

[2] https://github.com/weiaicunzai/pytorch-cifar100

[3] https://github.com/rwightman/pytorch-image-models

- ResNeXt-WSL (Mahajan et al., 2018): We use the Instagram pre-trained and ImageNet-1k fine-tuned model provided by Meta's research group[4]. Specifically, we use a model with ResNeXt101 32x8d architecture.
- ViT-Base (Dosovitskiy et al., 2020b): We use the ImageNet-21k pre-trained and ImageNet-1k fine-tuned model provided by the timm library[3]. Specifically, we use the ViT-Base that expects an input image size of 224 and have 16 patch-embeddings.

C. Post-hoc methods

C.1. Density-Aware Calibration (DAC)

Table 6. Layers from classifiers used for DAC. For neural networks that have block structures, we pick the very last layer of each block (denoted by Block-i for i_{th} block), and, additionally, the immediate layer before the first block (denoted by Pre-block), the layer just before the fully-connected layers at the end of neural networks (denoted by Penultimate), and the logits layer (denoted by Logits). For VGG, we use feature vectors after every max-pooling layer where the resolution of the feature map changes (denoted by Maxpool-i).

Dataset	Model	Used layers
CIFAR10	ResNet18	Pre-block, Block-1, ..., Block-4, Logits
	VGG16	Maxpool-1, ..., Maxpool-5, logits
	DenseNet121	Pre-block, Block-1, ..., Block-3, Penultimate, logits
CIFAR100	ResNet18	Pre-block, Block-1, ..., Block-4, Logits
	VGG16	Maxpool-1, ..., Maxpool-5, logits
	DenseNet121	Pre-block, Block-1, ..., Block-3, Penultimate, logits
ImageNet	ResNet152	Pre-block, Block-1, ..., Block-4, Logits
	DenseNet169	Pre-block, Block-1, ..., Block-3, Penultimate, logits
	Xception	Pre-block, Block-1, ..., Block-12, Penultimate, logits
	BiT-M (ResNetV2)	Pre-block, Block-1, ..., Block-34, Penultimate, logits
	ResNeXt-WSL (ResNeXt101 32x8d)	Pre-block, Block-1, ..., Block-4, logits
	ViT-Base	Pre-block, Block-1, ..., Block-24, Penultimate, logits

Even though DAC is capable of using every layer in a classifier due to its weighting scheme, we opt for a much simpler and faster version that uses a subset of layers. We follow a structured approach for choosing layers to end up with a well-distributed subset of layers: We choose (1) the last layer of each "block" in a neural network, e.g., ResNet or Transformer block, or, (2) the layer where the resolution or channel size of the feature vector changes, e.g., VGG. The intuition behind this approach is based on the fact that layers in neural networks represent an image at increasing levels of abstraction from low-level to high-level representation, and thus, different blocks or layers with different resolutions are expected to have different levels of representation. The approach of choosing a subset of layers to obtain different levels of representations has been applied to a wide range of works in the computer vision field (Zhang et al., 2018; Gatys et al., 2016; Chen & Koltun, 2017): Zhang et al. (2018) leverage different levels of representations for better measurement of the perceptual difference between two images; Gatys et al. (2016) leverage different levels of representations to separate image content from style for image style transfer task; Chen et al. (2017) applied the strategy to image synthesis task.

Based on the above idea, we select layers for each classifier as described in Tab. 6. For neural networks that have block structures, we pick the very last layers of each block (i.e., block-i for i_{th} block), and, additionally, the immediate layer before the first block (i.e., pre-block), e.g., the first max-pooling layer in ResNet, the layer just before the fully-connected layers at the end of neural networks (i.e., Penultimate), and the logits layer (i.e., Logits). For VGG, which does not have a block structure, we choose 5 different resolutions of layers, similar to Gatys et al. (2016) and Chen et al. (2017): Specifically, we use feature vectors after every max-pooling layer where the resolution of the feature map changes (i.e., Maxpool-i).

Note that the extracted feature vector from each layer is always converted to a single-dimensional vector by a pooling operation. For convolutional neural networks, we do spacial avg-pooling for each layer output. For Transformer architectures, we do avg-pooling w.r.t. token length.

[4] https://github.com/facebookresearch/WSL-Images

C.2. Baseline Post-hoc Methods

Here, we describe the details of the implementation of the baseline post-hoc methods.

- Temperature scaling (TS) (Guo et al., 2017): We use the implementation from the GitHub repository[5] provided by Zhang et al. (Zhang et al., 2020).
- Ensemble Temperature scaling (ETS) (Zhang et al., 2020): Ensemble version of TS with 4 parameters. We use the official implementation from their GitHub repository[5].
- Isotonic Regression for multi-class (IR) (Zhang et al., 2020): Decomposes the problem as one-versus-all problem to extend Isotonic Regression for multi-class setting. We use the official implementation from their GitHub repository[5].
- Accuracy preserving version of Isotonic Regression (IRM) (Zhang et al., 2020): We use the official implementation from their GitHub repository[5].
- Parameterized Temperature scaling (PTS) (Tomani et al., 2022): Sample-wise version of TS. Following their paper, PTS was trained as a neural network with 2 fully connected hidden layers with 5 nodes each, using a learning rate of 0.00005, batch size of 1000, and step size of 100,000. The top 10 most confident predictions were used as input.
- Dirichlet calibration (DIR) (Milios et al., 2018): Matrix scaling with off-diagonal and intercept regularization. Among their variants, we use MS-ODIR, which is intended for calibrating logits rather than probabilities and is claimed as the best-performing variant by the authors. We use the official implementation from their GitHub repository[6]. However, we encountered some difficulties and could not adapt the code to run ImageNet experiments.
- Intra-order preserving calibration (DIA) (Rahimi et al., 2020b): Among their variants, we use the method DIAG (diagonal intra-order-preserving), which works the best on average for various datasets and classifiers. We use their official implementation [7]. DIA can be run with or without hyperparameter optimization; for a fair comparison, since our other baselines, including DAC, have fixed hyperparameters, we show results without hyperparameter optimization.
- Calibration using Splines (SPL) (Gupta et al., 2021): We use their official implementation [8]. Following their paper, we use the natural cubic spline fitting method with 6 knots for all our experiments.

C.3. Training and Evaluation of DAC

Our post-hoc calibration method DAC does not require any GPU for the training phase as well as the inference phase. For computing s_l via the k-nearest neighbor method (equation (7)), we use the Faiss library[9] (Johnson et al., 2019) for efficient and fast similarity search, which can be run with a CPU or GPU. To minimize the loss function described in equation (9), we use Scipy optimization using a CPU.

D. Results for Additional Baseline-Calibration Methods

In addition to the state-of-the-art post-hoc calibration method which we compared in the main text of the paper, we show another 3 commonly used methods in this section and combine them with our proposed DAC method. Across all methods, classifiers, and datasets we see consistent improvements in post-hoc methods combined with DAC for macro-averaged ECE (calculated across all corruptions and levels of severity) in Tab. 7. Moreover, in Tab. 8 we show the additional gain of the model and in Fig. 7 we demonstrate how ECE behaves across different severity levels of corruption.

[5] https://github.com/zhang64-llnl/Mix-n-Match-Calibration

[6] https://github.com/dirichletcal/experiments_dnn

[7] https://github.com/AmirooR/IntraOrderPreservingCalibration

[8] https://github.com/kartikgupta-at-anu/spline-calibration

[9] https://github.com/facebookresearch/faiss

Table 7. **ECE ($\times 10^2$) for Additional calibration post-hoc methods:** Mean expected calibration error across all test domain-shift scenarios. For each model, the macro-averaged ECE (with equal-width binning and 15 bins) is computed across all corruptions from severity=0 (in-domain) until severity=5 (heavily corrupted). Also for these additional post-hoc calibration methods results are consistently better calibrated when paired with our method. (lower ECE is better)

	UNCAL	BASELINE CALIBRATION METHODS			COMBINATION WITH DAC (OURS)		
	-	PTS	IR	DIR	PTS	IR	DIR
C10 RESNET18	19.27	4.99	6.27	5.55	**4.55**	5.42	4.80
C10 VGG16	19.05	6.74	7.92	8.48	**5.66**	6.91	7.62
C10 DENSENET121	19.26	4.75	6.61	7.47	**4.64**	5.88	6.38
C100 RESNET18	16.44	11.48	14.40	12.02	10.81	12.04	**10.58**
C100 VGG16	34.41	9.78	15.31	15.52	**7.59**	10.50	8.97
C100 DENSENET121	23.83	10.67	13.74	13.02	**10.53**	12.11	11.77
IMG RESNET152	10.50	7.32	12.41	-	**4.78**	9.91	-
IMG DENSENET169	13.28	7.93	15.05	-	**6.98**	12.94	-
IMG XCEPTION	30.49	9.15	17.94	-	**9.13**	14.29	-
IMG BIT-M	11.71	6.50	13.10	-	**5.49**	10.37	-
IMG RESNEXT-WSL	15.44	7.21	12.53	-	**6.27**	9.26	-
IMG VIT-B	**3.78**	5.20	11.85	-	3.80	11.49	-

Table 8. **Deltas for Additional post-hoc methods:** Difference in expected calibration error ($\times 10^2$) of post-hoc calibration methods with and without our method DAC. InD: In-domain, $SEV.5$: Heavily corrupted (severity of 5) and ALL: Macro-averaged ECE across all corruptions from severity=0 (in-domain) until severity=5 (heavily corrupted). For ALL, we additionally report the ratio to indicate overall performance gain. (negative deltas are better)

	IND	SEV. 5	ALL	
PTS	-0.37	-0.94	-0.43	(9.8%)
IR	-0.17	-1.44	-0.74	(13.4%)
DIR	-0.56	-1.44	-0.72	(14.8%)

A.) CIFAR10 RESNET18

	IND	SEV. 5	ALL	
PTS	+0.45	-3.83	-1.77	(21.0%)
IR	-1.58	-6.45	-4.30	(31.5%)
DIR	-1.77	-9.22	-5.80	(42.3%)

B.) CIFAR100 VGG16

	IND	SEV. 5	ALL	
PTS	-0.14	-1.79	-0.82	(12.0%)
IR	-0.97	-2.66	-1.93	(14.1%)

C.) IMAGENET DENSENET169

	IND	SEV. 5	ALL	
PTS	+0.06	-1.87	-0.84	(15.1%)
IR	-0.73	-4.24	-2.41	(20.2%)

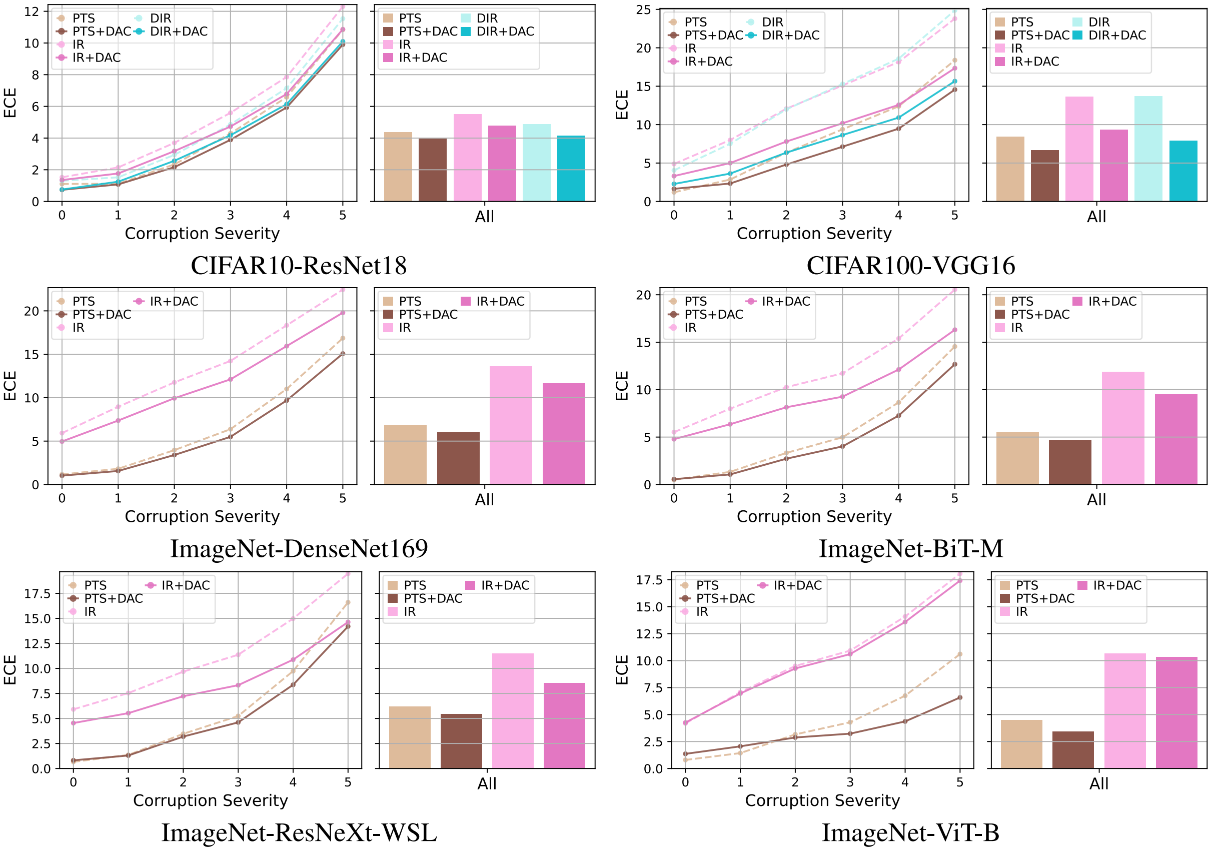
D.) IMAGENET BIT-M

	IND	SEV. 5	ALL	
PTS	+0.15	-2.40	-0.76	(12.4%)
IR	-1.37	-4.84	-2.97	(25.8%)

E.) IMAGENET RESNEXT-WSL

	IND	SEV. 5	ALL	
PTS	+0.57	-4.02	-1.09	(24.2%)
IR	+0.08	-0.64	-0.29	(2.7%)

F.) IMAGENET VIT-B



CIFAR10-ResNet18

CIFAR100-VGG16

ImageNet-DenseNet169

ImageNet-BiT-M

ImageNet-ResNeXt-WSL

ImageNet-ViT-B

Figure 7. **ECE for additional post-hoc methods:** Expected calibration error ($\times 10^2$) of post-hoc methods with and without our method DAC for different model and dataset combinations. **Line plots**: Macro-averaged ECE across all corruption types shown for each corruption severity, from in-domain to OOD. **Bar plots**: Macro-averaged ECE across all corruption types as well as across all severities (lower ECE is better).

E. Results for Additional Calibration Measures

Even though ECE is most commonly used for evaluating calibration performance, other metrics have been proposed as well. Here we want to show that our results on ECE with equal-width binning are consistent with various other calibration methods. We additionally evaluate based on: **a.** ECE using equal-mass binning (Tab. 9), **b.** ECE based on kernel density estimation (ECE-KDE) (Zhang et al., 2020) (Tab. 10) and **c.** class-wise ECE (Kull et al., 2019) (Tab. 11). Moreover, we show that negative log-likelihood (Tab. 12) is kept similar for our method compared to baseline calibration methods without DAC. In each of these tables, we report macro-averaged ECE or NLL scores. For each model, we compute the macro-averaged ECE or NLL across all corruptions from severity=0 (in-domain) until severity=5 (OOD). For spline authors didn't provide ways to calibrate the probabilistic predictions; that is, why we were not able to calculate the negative log-likelihood, and class-wise ECE for this method.

Table 9. Mean **ECE** ($\times 10^2$) **with equal-mass binning** (15 bins) computed across all corruptions from severity=0 (in-domain) until severity=5 (heavily corrupted). (lower ECE is better).

	Uncal	Baseline Calibration Methods					Combination with DAC (Ours)				
	-	TS	ETS	IRM	DIA	SPL	TS	ETS	IRM	DIA	SPL
C10 ResNet18	19.26	4.93	4.93	5.91	6.45	6.26	4.45	**4.37**	4.81	4.60	4.77
C10 VGG16	19.05	6.31	6.36	7.44	9.88	7.52	**5.60**	**5.60**	6.18	6.33	5.68
C10 DenseNet121	19.26	5.20	5.20	6.61	7.84	6.59	4.57	4.57	5.63	6.62	**4.42**
C100 ResNet18	16.41	11.37	10.72	12.25	9.24	10.39	10.65	9.11	10.01	**8.45**	9.75
C100 VGG16	34.40	11.57	11.57	13.21	14.69	10.75	6.49	**6.48**	8.06	10.30	7.48
C100 DenseNet121	23.83	8.78	8.76	12.03	11.02	9.72	8.74	**8.40**	9.96	15.07	9.17
IMG ResNet152	10.49	4.47	4.05	5.18	7.16	5.56	3.48	**3.37**	3.49	3.64	3.65
IMG DenseNet169	13.27	6.58	6.39	7.36	8.43	7.11	4.80	**3.91**	4.50	6.29	4.60
IMG Xception	30.48	8.83	8.42	12.93	9.83	10.79	8.81	**8.00**	8.37	8.99	8.49
IMG BiT-M	11.71	7.15	6.54	6.92	7.46	6.60	4.40	3.99	4.19	5.52	**3.75**
IMG ResNeXt-WSL	15.40	8.01	8.01	8.03	8.31	6.17	7.29	5.91	5.73	6.30	**3.97**
IMG ViT-B	3.77	4.22	3.76	4.21	5.84	3.94	3.79	**3.39**	3.96	5.51	3.58

Table 10. Mean **ECE-KDE** ($\times 10^2$) computed across all corruptions from severity=0 (in-domain) until severity=5 (heavily corrupted). (lower ECE-KDE is better).

	Uncal	Baseline Calibration Methods					Combination with DAC (Ours)				
	-	TS	ETS	IRM	DIA	SPL	TS	ETS	IRM	DIA	SPL
C10 ResNet18	18.63	4.93	4.93	5.78	6.34	6.27	4.35	**4.30**	4.67	4.58	4.75
C10 VGG16	18.44	6.25	6.30	7.15	9.74	7.53	**5.56**	**5.56**	5.94	6.31	5.67
C10 DenseNet121	18.64	5.17	5.17	6.30	7.61	6.51	4.51	4.51	5.20	6.47	**4.35**
C100 ResNet18	15.99	11.08	10.55	11.93	9.13	10.37	10.39	9.05	9.69	**8.33**	9.67
C100 VGG16	33.83	11.54	11.54	13.14	14.46	10.80	6.50	**6.49**	7.81	10.14	7.37
C100 DenseNet121	23.26	8.79	8.83	11.66	10.99	9.79	8.74	**8.47**	9.58	14.86	9.15
IMG ResNet152	10.28	4.44	4.16	5.11	7.07	5.66	**3.49**	3.55	3.52	3.72	3.71
IMG DenseNet169	13.04	6.49	6.39	7.25	8.32	7.20	4.75	**4.01**	4.43	6.21	4.66
IMG Xception	30.18	8.89	8.54	12.93	9.84	10.93	8.87	**8.15**	8.35	9.01	8.53
IMG BiT-M	11.60	7.14	6.59	6.86	7.45	6.66	4.44	4.10	4.13	5.51	**3.80**
IMG ResNeXt-WSL	14.90	7.55	7.55	7.74	8.07	6.32	6.82	5.66	5.43	6.13	**4.01**
IMG ViT-B	3.72	4.15	3.78	4.07	5.75	3.96	3.74	**3.42**	3.82	5.44	3.63

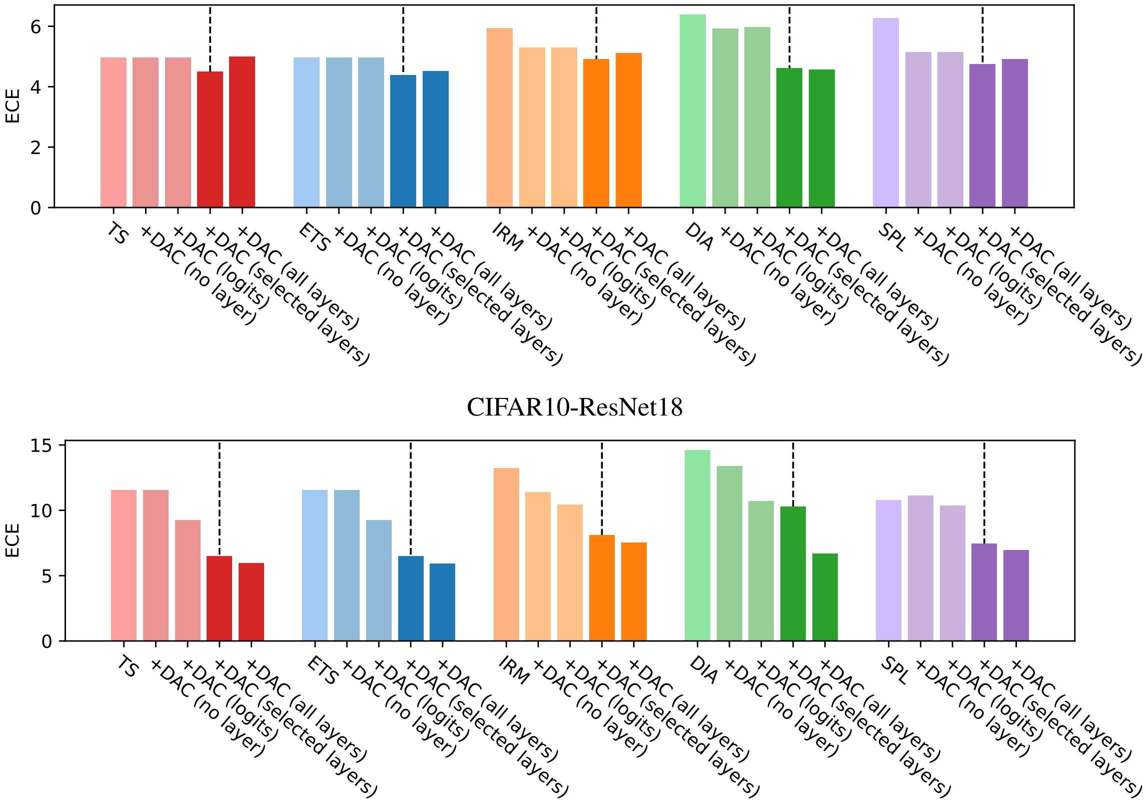
Table 11. Mean **Class-wise ECE** ($\times 10^2$) computed across all corruptions from severity=0 (in-domain) until severity=5 (heavily corrupted). (lower class-wise ECE is better). Note that since SPL only calibrates the highest predicted confidence, it is not directly possible to evaluate class-wise ECE.

	Uncal	Baseline Calibration Methods				Combination with DAC (Ours)			
	-	TS	ETS	IRM	DIA	TS	ETS	IRM	DIA
C10 ResNet18	41.33	22.96	22.96	22.94	24.46	22.45	**22.43**	22.84	22.66
C10 VGG16	40.72	24.44	24.46	24.80	28.41	**24.07**	**24.07**	24.37	24.68
C10 DenseNet121	41.49	24.70	24.70	24.63	25.82	**24.04**	**24.04**	24.20	24.86
C100 ResNet18	54.32	48.89	48.42	54.09	48.44	48.35	46.73	52.22	**46.53**
C100 VGG16	80.43	47.73	47.73	48.30	52.59	**42.11**	**42.11**	45.63	46.22
C100 DenseNet121	61.97	45.78	45.82	51.99	48.94	45.70	**45.56**	51.74	56.51
IMG ResNet152	58.49	54.61	54.25	62.23	58.77	52.76	**52.73**	54.18	55.49
IMG DenseNet169	61.05	56.71	56.29	61.32	59.90	55.37	**54.68**	56.54	58.20
IMG Xception	84.33	54.22	53.93	80.07	64.24	54.19	**53.38**	61.96	63.65
IMG BiT-M	55.78	53.45	53.05	58.28	56.64	50.63	**50.44**	54.16	55.11
IMG ResNeXt-WSL	46.54	38.87	38.87	47.64	45.08	**37.60**	37.77	40.40	43.00
IMG ViT-B	49.95	50.10	49.91	51.23	53.36	49.96	**49.78**	51.35	53.17

Table 12. Mean **negative log-likelihood** computed across all corruptions from severity=0 (in-domain) until severity=5 (heavily corrupted). Note that since SPL only calibrates the highest predicted confidence, it is not directly possible to evaluate negative log-likelihood.

	Uncal	Baseline Calibration Methods				Combination with DAC (Ours)			
	-	TS	ETS	IRM	DIA	TS	ETS	IRM	DIA
C10 ResNet18	1.4496	0.8713	0.8713	0.8718	0.9227	0.8666	0.8657	0.8883	0.8675
C10 VGG16	1.4500	0.8036	0.8040	0.8008	0.8912	0.8029	0.8029	0.8065	0.8068
C10 DenseNet121	1.3284	0.8918	0.8918	0.8905	0.9045	0.8827	0.8827	0.9000	0.8868
C100 ResNet18	2.4690	2.3330	2.3088	2.3575	2.2879	2.3356	2.2830	2.3159	2.2721
C100 VGG16	3.6090	2.4193	2.4193	2.4340	2.5827	2.3232	2.3231	2.3585	2.3899
C100 DenseNet121	2.4672	2.1404	2.1427	2.1535	2.1400	2.1417	2.1413	2.1890	2.2775
IMG ResNet152	2.6844	2.5870	2.5887	2.4654	2.6669	2.5691	2.5772	2.5611	2.5763
IMG DenseNet169	2.7371	2.5975	2.6006	2.5211	2.6745	2.5677	2.5650	2.5565	2.6126
IMG Xception	4.2438	3.3895	3.3977	3.2663	3.4935	3.3893	3.3915	3.4136	3.4733
IMG BiT-M	2.3315	2.2874	2.2914	2.2256	2.3145	2.2689	2.2775	2.2363	2.2588
IMG ResNeXt-WSL	2.1246	1.9379	1.9379	1.7881	1.9260	1.9407	1.9370	1.8354	1.8381
IMG ViT-B	2.2560	2.2585	2.2617	2.3009	2.3107	2.2561	2.2597	2.2900	2.3044

F. Ablation Study: How Information from Hidden Layers Benefits Calibration with DAC



CIFAR10-ResNet18

CIFAR100-VGG16

Figure 8. Ablation study on how information from hidden layers benefits the calibration performance of DAC. The y-axis shows the macro-averaged ECE ($\times 10^2$) computed across all corruptions from severity=0 (in-domain) until severity=5 (heavily corrupted). DAC boosts calibration performance significantly as soon as information from hidden layers is available (”selected layers” and ”all layers”) compared to DAC with access to only the ”logits layer”.

In order to investigate the importance of hidden layers from classifiers towards the calibration performance of DAC, we conduct an ablation study. In this study, we analyze the sensitivity of ECE with regard to different subsets of hidden layers that our DAC has access to.

We compare the calibration performance of the following variations:

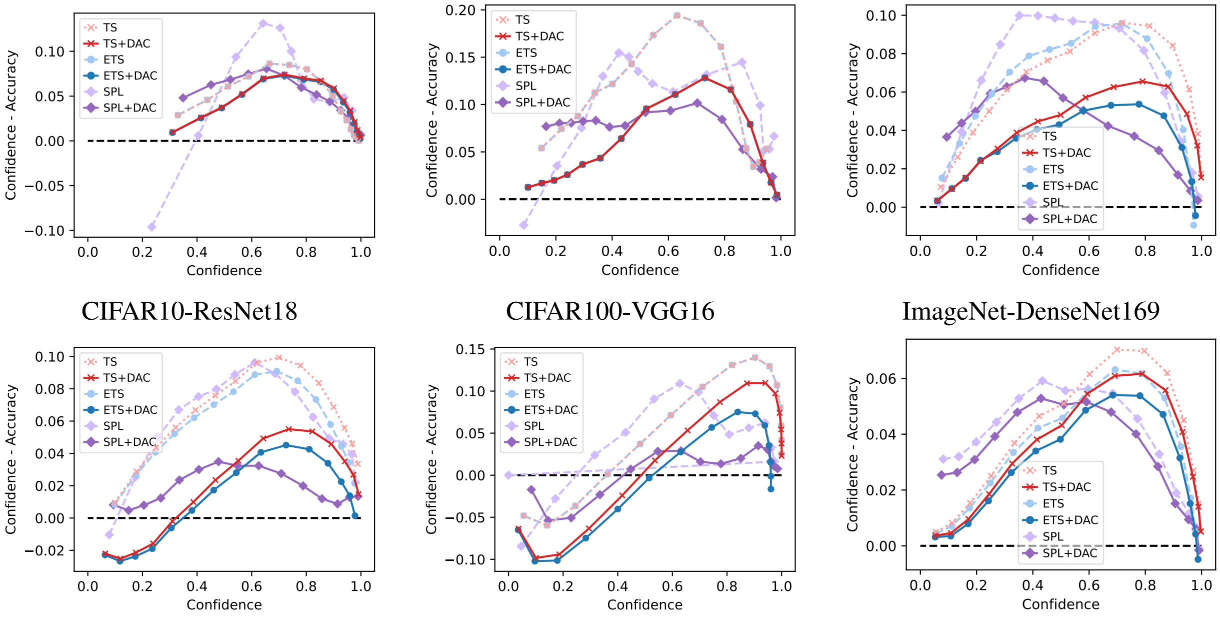
- **w/o DAC**: Baseline post-hoc calibration method without DAC.
- **DAC (no layer)**: Baseline method + DAC using no layer at all. If DAC has no access to any layer in the classifier, only the bias term w_0 remains in equation (8), thus in this setting DAC is equivalent to temperature scaling.
- **DAC (logits layer)**: Base method + DAC using only the logits layer.
- **DAC (selected layers)**: Base method + DAC using selected layers (a well distributed subset of layers throughout the classifier). This is the default DAC used throughout the paper.
- **DAC (all layers)**: Base method + DAC using all layers of the classifiers.

Fig. 8 shows macro-averaged ECE across all corruptions from severity=0 (in-domain) until severity=5 (OOD). The results indicate that DAC using solely logits layer can already boost calibration compared to the respective stand-alone baseline methods. However, purely relying on the logits layer as a basis for post-hoc calibration is still suboptimal. DAC boosts the calibration performance significantly as soon as information from hidden layers is available to DAC. However, by increasing the access further from selected layers to all layers, an increase in calibration performance can not be observed in all scenarios. This indicates that more layers can benefit DAC's performance; however, we assume that too many layers can also cause an overfitting issue due to having too many parameters to optimize.

G. Reliability Diagrams

Reliability diagrams allow insights into calibration performance by showing the difference between a method's confidence in its predictions and its accuracy (Guo et al., 2017). Following Maddox et al. (Maddox et al., 2019), we show the following reliability diagrams. We split the test data into 15 bins based on the confidence values, so that each bin contains a uniform number of data points. Then, we evaluate the accuracy and mean confidence for each bin. For a well-calibrated model, this difference should be close to zero for each bin.

In Fig. 9, we show the reliability diagrams for 6 different dataset-classifier pairs on CIFAR-C or ImageNet-C dataset, across all corruptions and severity levels combined in one reliability diagram. The figure shows that our post-hoc calibration method DAC successfully boosts the existing post-hoc methods' calibration performance in domain-shift scenarios.



CIFAR10-ResNet18 CIFAR100-VGG16 ImageNet-DenseNet169

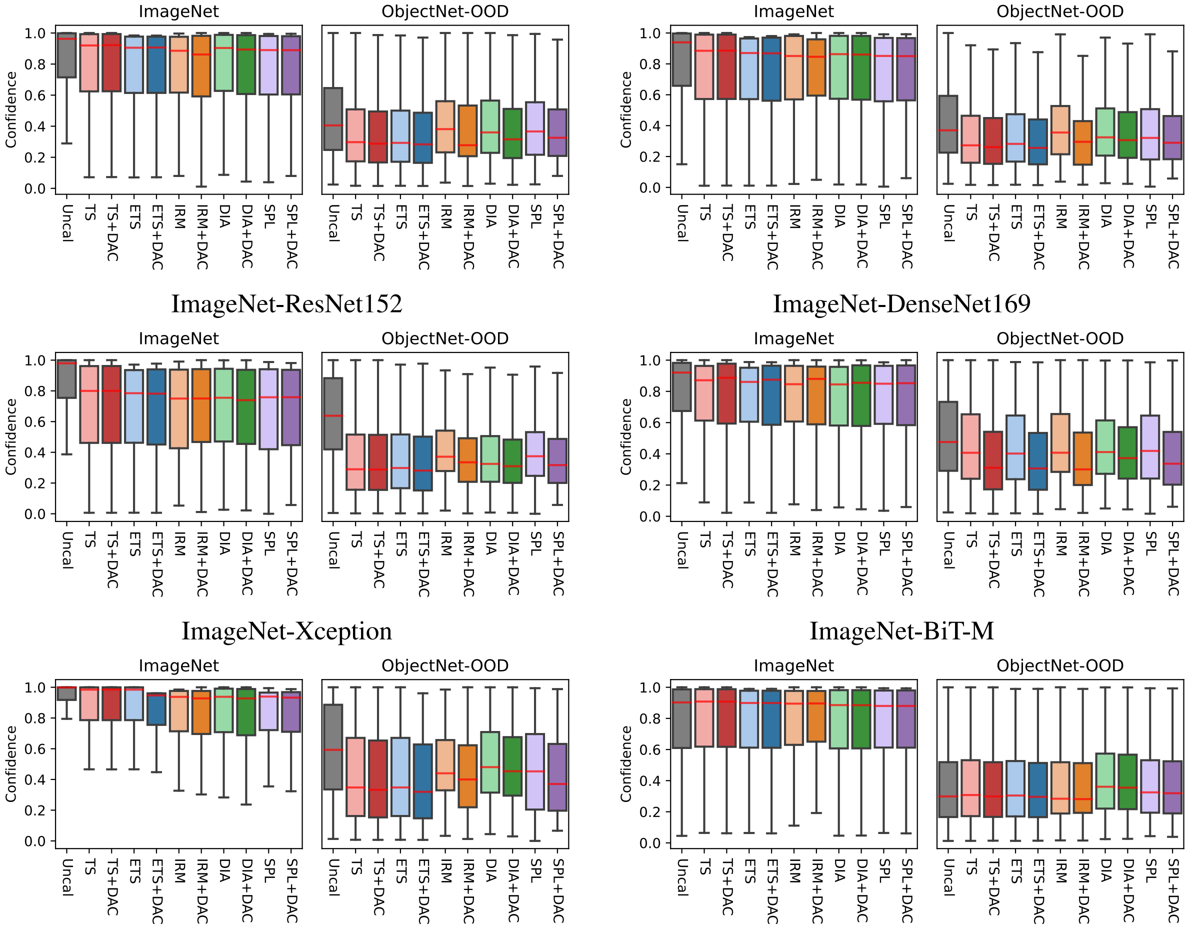
ImageNet-BiT-M ImageNet-ResNeXt-WSL ImageNet-ViT-B

Figure 9. Reliability Diagrams for all corruptions and severity levels combined (with equal mass-binning for 15 bins).

H. Additional Results: OOD Scenarios

Here, we include additional results from the OOD experiments. Fig. 10 summarizes the in-domain/OOD confidence distributions in various cases, where we see that DAC can in general better separate the in-domain data from OOD data, which is desirable. Tab. 13 provides a more comprehensive summary of the quantitative OOD results for various network backbones. We observe consistent improvement with DAC in majority of the cases.

Note that BiT-M, ResNeXt-WSL, and ViT-B are pre-trained with additional data, as mentioned in Section 4. This might influence the outcome and the validity of the OOD experimental setup. The effect of backbone pre-training on the OOD tasks could be an interesting topic to investigate for future work.



ImageNet-ResNet152

ImageNet-DenseNet169

ImageNet-Xception

ImageNet-BiT-M

ImageNet-ResNeXt-WSL

ImageNet-ViT-B

Figure 10. Boxplots of confidence values for ImageNet test set, and ObjectNet-OOD dataset. By combining the existing calibration methods with our DAC method, all classifiers output lower confidence values for novel-class images.

Table 13. Additional OOD results using ImageNet-1k/ObjectNet-OOD as in- domain/OOD test set, respectively.

	FPR @95% ↓	Det. Err ↓	AU-ROC ↑	AUPR-In ↑	AUPR-Out ↑
TS	21.52	23.18	84.75	80.41	86.83
+DAC	**21.03**	**22.70**	**85.27**	**81.20**	**87.30**
ETS	21.52	23.18	84.75	**80.41**	86.83
+DAC	**21.36**	**22.70**	**85.24**	80.11	**87.29**
IRM	16.10	21.94	83.55	80.37	86.32
+DAC	**14.90**	**21.56**	**85.19**	**80.75**	**87.25**
DIA	22.15	24.61	83.38	79.07	85.65
+DAC	**21.06**	**23.52**	**84.49**	**80.34**	**86.64**
SPL	21.70	24.54	83.45	79.29	85.72
+DAC	**21.13**	**22.69**	**85.23**	**81.25**	**87.22**

a.) ResNet152

	FPR @95% ↓	Det. Err ↓	AU-ROC ↑	AUPR-In ↑	AUPR-Out ↑
TS	21.47	22.86	84.92	81.14	86.87
+DAC	**20.73**	**22.30**	**85.50**	**81.86**	**87.46**
ETS	21.92	23.15	84.72	79.82	86.72
+DAC	**20.90**	**22.29**	**85.51**	**82.16**	**87.47**
IRM	21.51	22.31	83.62	81.22	86.10
+DAC	**4.87**	**20.72**	**85.81**	**82.73**	**87.90**
DIA	22.09	24.26	83.67	79.86	85.77
+DAC	**21.36**	**23.64**	**84.35**	**80.63**	**86.47**
SPL	21.66	24.38	83.65	79.95	85.78
+DAC	**20.89**	**22.30**	**85.54**	**82.05**	**87.56**

b.) DenseNet169

	FPR @95% ↓	Det. Err ↓	AU-ROC ↑	AUPR-In ↑	AUPR-Out ↑
TS	28.89	28.57	77.83	71.57	79.28
+DAC	**28.71**	**28.51**	**77.91**	**71.71**	**79.35**
ETS	28.87	28.66	77.75	71.53	79.23
+DAC	**28.67**	**28.50**	**77.91**	**71.79**	**79.35**
IRM	19.95	28.85	74.82	71.89	77.33
+DAC	**15.70**	**27.58**	**78.09**	**73.02**	**79.51**
DIA	28.36	28.09	78.84	74.98	80.09
+DAC	**28.27**	**27.82**	**79.04**	**74.96**	**80.25**
SPL	30.90	30.83	75.36	70.72	77.21
+DAC	**28.71**	**28.50**	**77.90**	**71.56**	**79.31**

c.) Xception

	FPR @95% ↓	Det. Err ↓	AU-ROC ↑	AUPR-In ↑	AUPR-Out ↑
TS	26.42	28.59	78.47	70.72	81.88
+DAC	**23.68**	**25.23**	**82.79**	**78.77**	**84.99**
ETS	26.42	28.59	78.47	70.72	81.88
+DAC	**23.69**	**25.21**	**82.82**	**78.96**	**85.00**
IRM	33.44	27.43	77.73	72.45	81.79
+DAC	**9.14**	**23.54**	**82.84**	**79.41**	**85.19**
DIA	26.67	28.26	79.39	74.55	82.06
+DAC	**25.32**	**26.58**	**81.40**	**77.92**	**83.64**
SPL	26.66	29.01	78.07	70.84	81.45
+DAC	**23.64**	**25.25**	**82.81**	**79.31**	**84.96**

d.) BiT-M

	FPR @95% ↓	Det. Err ↓	AU-ROC ↑	AUPR-In ↑	AUPR-Out ↑
TS	22.52	25.47	82.30	75.30	85.83
+DAC	**22.05**	**25.05**	**82.83**	**76.00**	**86.26**
ETS	**22.52**	25.47	82.30	75.30	85.83
+DAC	23.15	**25.05**	**83.07**	**78.77**	**86.33**
IRM	23.20	25.61	80.66	**77.06**	84.80
+DAC	**10.86**	**23.50**	**82.75**	76.13	**86.06**
DIA	25.19	27.96	79.37	72.47	82.84
+DAC	**24.77**	**27.48**	**79.94**	**73.10**	**83.37**
SPL	28.23	26.69	80.84	**76.46**	84.17
+DAC	**22.63**	**25.04**	**82.84**	75.44	**86.29**

e.) ResNeXt-WSL

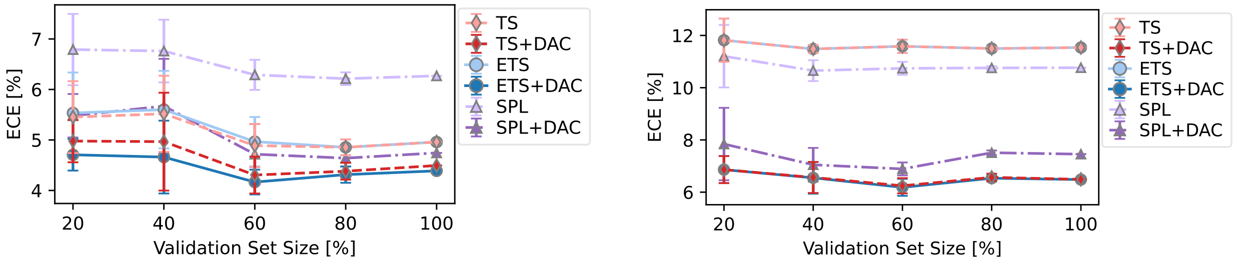
	FPR @95% ↓	Det. Err ↓	AU-ROC ↑	AUPR-In ↑	AUPR-Out ↑
TS	22.52	24.16	83.77	79.95	85.54
+DAC	**22.15**	**23.75**	**84.24**	**80.74**	**85.95**
ETS	22.52	24.16	83.77	79.95	85.54
+DAC	**22.18**	**23.76**	**84.22**	**80.42**	**85.94**
IRM	20.76	24.01	83.69	79.31	85.30
+DAC	**5.42**	**21.22**	**83.87**	**80.46**	**85.48**
DIA	23.53	25.73	82.22	78.12	84.50
+DAC	**23.28**	**25.50**	**82.53**	**78.63**	**84.77**
SPL	**21.55**	23.98	83.90	80.09	85.63
+DAC	22.12	**23.73**	**84.21**	**80.77**	**85.88**

f.) ViT-B

I. Data Efficiency Analysis

I.1. Additional Results

We show additional data efficiency diagrams for models on CIFAR10 and CIFAR100. As for ImageNet in the main text, we encounter no drastic sensitivity to validation set size. We conclude that these post-hoc methods combined with DAC can be trained on very small validation set sizes.



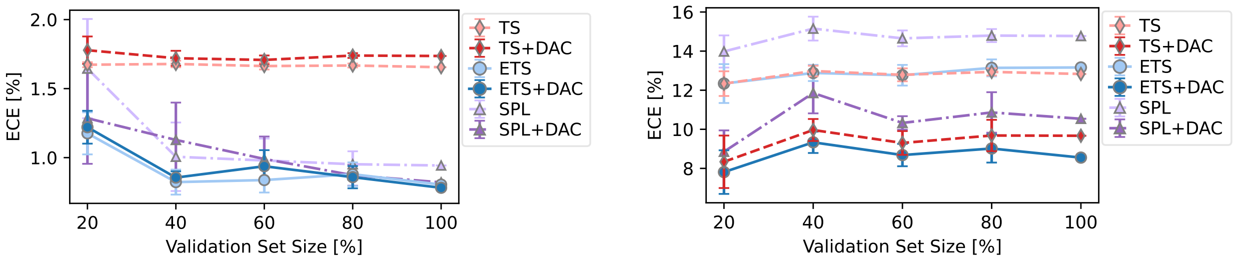
CIFAR10-ResNet18

CIFAR100-VGG16

Figure 11. Data efficiency diagrams for macro-averaged ECE ($\times 10^2$) (with 15 bins across all corruptions from severity=0 (in-domain) until severity=5 (heavily corrupted)) from 10% to 100% validation set size.

I.2. Trade-off between in-domain and domain-shift performance

We observe that the larger the validation set gets, the more likely the calibration method is to overfit to in-domain data, leading to a degradation in performance on domain shift and OOD data. In Figure 12, we show the data efficiency diagrams for ECE ($\times 10^2$) with 15 bins from 10% to 100% validation set size, for the in-domain case and the case of corruption severity=5 (heavily corrupted), respectively. We show that while the in-domain calibration performance benefits from the size of the validation set, the domain-shift calibration performance is worsen due to overfitting. Fig. 6 in the main text shows the mean ECE across all test-domain shift scenarios, including in-domain, and thus incorporates this overfitting behavior on large validation set sizes.



IMG-DenseNet169 (In-domain)

IMG-DenseNet169 (Sev. 5)

Figure 12. Data efficiency diagrams for ECE ($\times 10^2$) with 15 bins from 10% to 100% validation set size. Left: image corruption severity=0 (in-domain). Right: image corruption severity=5 (heavily corrupted)). The figures show the trade-off between in-domain and out-of-domain calibration errors of DAC.

J. Computational Cost

In this section, we compare the computational cost of DAC and the existing methods. In short, the computational cost of DAC is similar to that of existing post-hoc methods during both the training and inference phases.

We first compare the training speed of DAC, DIA, ETS, and SPL for DenseNet169 trained on ImageNet, using an NVIDIA Titan X (12GB) GPU. The table below shows that DAC achieves a total training time of only about 14 minutes. Compared to the overall training process of the classifier, which takes at least several hours, the additional computational cost added by DAC is minor.

	Total Training time (s)	1. Extract features for train set (s)	2. Extract features/logits for validation set (s)	3. Optimization (s)
DAC	825.98	175.72	452.91 (include KNN)	197.35
DIA	11090.09	-	130.49	10959.60
ETS	149.35	-	130.49	18.86
SPL	131.04	-	130.49	0.55

Table 14. Training time comparison: Our proposed method DAC vs. existing post-hoc calibration methods (DIA, ETS, SPL).

In addition, we compare the per-sample inference speed. As shown in the table below, we found that the inference speed of DenseNet169 combined with DAC is comparable to the existing calibration methods. This can be attributed to the efficient GPU-accelerated KNN search in DAC, enabled by the Faiss library [10] (Johnson et al., 2019).

	Total per-sample inference time (ms)
DAC	37.12 (+-0.77)
DIA	38.23 (+-0.97)
ETS	34.05 (+-0.96)
SPL	34.00 (+-0.96)

Table 15. Per-sample inference time comparison: Our proposed method DAC vs. existing post-hoc calibration methods (DIA, ETS, SPL).

[10] https://github.com/facebookresearch/faiss