Supplementary Material for
Post-hoc Uncertainty Calibration for Domain Drift Scenarios

1. Summary

We provide further details on the implementation of our algorithm as well as additional results. This appendix is structured as follows.

- In section 2, we first formalize our algorithm in Algorithm 1. We then provide more details on the test perturbations used for our analyses, along with their parameter sets.
- In section 3, we report supplementary results, including additional metrics as well as data on additional baselines and additional experiments on the robustness of our findings.
- In section 4, we consolidate our results into a brief recommendation for practitioners.

2. Implementation details

2.1. Algorithm

Algorithm 1 Tuning of a post-calibration method for domain shift scenarios

Input: Classification model $Y = f(X)$, validation set (X, Y), number of perturbation levels N, number of classes C, initial parameter $\varepsilon_{\text{init}}$.

```
Compute min and max accuracy: $acc_{\min} = 1/C$; $acc_{\max} = \text{acc}(f(X), Y)$
Compute $N$ evenly spaced accuracy levels $A = \{acc_{\min}, acc_{\min} + \frac{acc_{\max} - acc_{\min}}{N-1}, \ldots, acc_{\max}\}$
Initialise empty perturbed validation set $(X_{\mathcal{E}}, Y_{\mathcal{E}})$
for $i$ in 1:N do
    if $i = 1$ then
        Set $\varepsilon_i = \varepsilon_{\text{init}}$
    end if
    Compute $X_{\varepsilon_i} = X + \mathcal{N}(0, \varepsilon_i)$, by drawing a sample from a Gaussian $\mathcal{N}(0, \varepsilon_i)$ with variance $\varepsilon_i$ for every pixel $(j, k)$ in every image $x_{j,k} \in X$
    Minimize $\text{acc}(f(X_{\varepsilon_i}), Y) - A_i$ with respect to $\varepsilon_i$, using a Nelder-Mead optimizer
    Compute $X_{\varepsilon_i} = X + \mathcal{N}(0, \varepsilon_i)$ using optimized $\varepsilon_i$
    Add $(X_{\varepsilon_i}, Y)$ to $(X_{\mathcal{E}}, Y_{\mathcal{E}})$
    if $i < N$ then
        Initialise $\varepsilon_{i+1} = \varepsilon_i$
    end if
end for
Tune post-processing method on $(X_{\mathcal{E}}, Y_{\mathcal{E}})$
```

2.2. Perturbation strategies

For the affine test perturbation strategies (Table 1) we chose 10 levels of perturbation with increasing perturbation strength until random levels of accuracy were reached (or parameters could not be increased any further). We started all test perturbation sequences at no perturbation and list specific levels of perturbation in Table 1.

For Imagenet corruptions, we follow [1] and report test accuracy as well as accuracy under maximum domain shift in Table 2.

Table 1: For rotation, perturbation is the (left or right) rotation angle in degrees, shift is measured in pixels in x or y direction, for shear the perturbation is measured as shear angle in counter-clockwise direction in degrees, for zoom the perturbation is zoom in x or y direction.

Perurbation	Perturbation-specific parameter									
rot left	0	350	340	330	320	310	300	290	280	270
rot right	0	10	20	30	40	50	60	70	80	90
shear	0	10	20	30	40	50	60	70	80	90
xyshift	0	2	4	6	8	10	12	14	16	18
xshift	0	2	4	6	8	10	12	14	16	18
xyshift	0	2	4	6	8	10	12	14	16	18
xyzoom	1	0.90	0.80	0.70	0.60	0.50	0.40	0.30	0.20	0.10
xzoom	1	0.90	0.80	0.70	0.60	0.50	0.40	0.30	0.20	0.10
yzoom	1	0.90	0.80	0.70	0.60	0.50	0.40	0.30	0.20	0.10

Table 2: Accuracies for Imagenet perturbations in-domain and with maximum shift.

Perurbation	Accuracy	
	In-Domain	Max Domain-Shift
shot noise	0.7452	0.07752
impulse noise	0.7452	0.07104
defocus blur	0.7452	0.14784
glass blur	0.7452	0.06904
motion blur	0.7452	0.09696
zoom blur	0.7452	0.22864
snow	0.7452	0.17776
frost	0.7452	0.25016
fog	0.7452	0.40912
brightness	0.7452	0.56776
contrast	0.7452	0.06416
elastic transform	0.7452	0.14480
pixelate	0.7452	0.19216
jpeg compression	0.7452	0.41136
gaussian blur	0.7452	0.10016
saturate	0.7452	0.47952
spatter	0.7452	0.30808
speckle noise	0.7452	0.18296

3. Additional results

3.1. Additional baselines

In addition to the state-of-the-art post-calibrators analysed in detail in the main paper, we also assessed the effect of tuning based on a perturbed validation set for additional baselines. Here, we report results for CIFAR-10 for Platt scaling [3], histogram binning [4] and a recently proposed approach combining Platt scaling with histogram binning (PBMC) [2]. Table 3 reveals that also these baselines benefit from tuning on a perturbed validation set; note however that overall ECE was consistently higher for these baselines compared to IR-P, for all architectures.

3.2. Additional metrics

In addition to the expected calibration error as reported in the main paper, we also compute a debiased ECE, recently proposed in [2], that can be more robust than the standard definition of ECE. Also with this measure, our approach improves all baselines consistently, with IRM-P, IR-P and TS-IR-P performing best (Table 4).

Table 3: Mean micro-average ECE across all affine test perturbations for the additional baselines.

	Base	PS	HB	PBMC	PS-P	HB-P	PBMC-P
CIFAR VGG19	0.323	0.173	0.254	0.211	**0.075**	0.086	0.101
CIFAR ResNet50	0.202	0.211	0.220	0.210	0.181	0.101	**0.099**
CIFAR Den.Net121	0.206	0.177	0.205	0.191	0.109	**0.096**	0.105
CIFAR Mob.NetV2	0.159	0.180	0.191	0.187	0.182	0.099	**0.098**

Table 4: Debiased ECE for all baselines for CIFAR-10 and Imagenet.

	Base	TS-P	ETS-P	TS-IR-P	IR-P	IRM-P
CIFAR VGG19	0.371	0.065	0.070	0.061	0.058	**0.054**
CIFAR ResNet50	0.221	0.099	0.110	0.101	0.101	**0.089**
CIFAR DenseNet121	0.230	0.162	0.148	0.118	**0.100**	0.141
CIFAR MobileNetv2	0.176	0.129	0.152	0.109	**0.089**	0.132
ImgNet ResNet50	0.144	0.058	0.047	**0.042**	**0.042**	0.050
ImgNet ResNet152	0.144	0.042	0.039	**0.034**	0.045	0.055
ImgNet VGG19	0.064	0.108	0.087	0.079	**0.034**	0.055
ImgNet Den.Net169	0.129	0.027	0.027	**0.030**	0.049	0.060
ImgNet Eff.NetB7	0.109	0.089	0.055	**0.042**	0.056	0.068
ImgNet Xception	0.235	0.072	0.038	**0.035**	0.119	0.122
ImgNet MobileNetv2	0.070	0.113	0.084	0.080	**0.053**	0.074

Furthermore, we also computed the negative log-likelihood as well as the Brier score for all post-calibrators. Again, our approach results in consistent improvements over the state-of-the-art also in terms of these metrics (Tables 5 and 6 and Figures 1 and 2).

Table 5: NLL for all baselines for CIFAR-10 and Imagenet

	Base	TS	ETS	TS-IR	IR	IRM	TS-P	ETS-P	TS-IR-P	IR-P	IRM-P
C-VGG19	2.49	1.49	1.47	1.86	1.88	1.55	1.37	1.37	1.47	1.47	1.38
C-ResNet50	1.78	1.69	1.68	2.42	2.42	1.86	1.48	1.48	2.20	1.90	1.47
C-Den.Net121	1.86	1.62	1.60	2.77	2.82	1.81	1.42	1.40	2.00	1.96	1.40
C-Mob.Netv2	1.66	1.64	1.62	2.86	2.88	1.93	1.47	1.49	2.08	2.00	1.48
I-ResNet50	2.81	2.65	2.67	8.00	7.97	2.67	2.65	2.64	2.92	2.93	2.65
I-ResNet152	2.49	2.33	2.34	7.22	7.18	2.35	2.32	2.33	2.67	2.71	2.33
I-VGG19	2.94	2.92	2.92	8.64	8.63	2.94	2.97	2.94	3.21	3.15	2.94
I-Den.Net169	2.48	2.35	2.35	7.33	7.31	2.37	2.34	2.34	2.74	2.80	2.35
I-Eff.NetB7	2.51	2.54	2.51	6.98	6.98	2.53	2.51	2.51	2.89	2.89	2.45
I-Xception	2.90	2.55	2.56	7.26	7.09	2.57	2.55	2.58	2.93	3.08	2.61
I-Mob.Netv2	3.45	3.58	3.51	10.3	10.2	3.67	3.52	3.48	3.66	3.62	3.51

Table 6: Brier score for all baselines for CIFAR-10 and Imagenet

	Base	TS	ETS	TS-IR	IR	IRM	TS-P	ETS-P	TS-IR-P	IR-P	IRM-P
C-VGG19	.731	.603	.600	.617	.620	.610	.565	.566	.574	.574	.566
C-ResNet50	.677	.663	.660	.681	.681	.664	.622	.624	.664	.659	.620
C-Den.Net121	.631	.600	.598	.618	.618	.601	.593	.587	.623	.607	.584
C-Mob.NetV2	.644	.640	.636	.656	.656	.639	.619	.627	.655	.642	.621
I-ResNet50	.667	.644	.648	.692	.692	.649	.646	.644	.645	.643	.644
I-ResNet152	.620	.597	.598	.645	.643	.600	.597	.596	.597	.597	.598
I-VGG19	.688	.686	.687	.732	.732	.687	.699	.694	.691	.681	.688
I-Den.Net169	.620	.602	.602	.650	.650	.604	.600	.600	.595	.596	.603
I-Eff.NetB7	.621	.634	.619	.635	.635	.608	.617	.612	.584	.586	.608
I-Xception	.682	.625	.621	.657	.661	.627	.624	.620	.611	.627	.635
I-Mob.NetV2	.745	.767	.758	.803	.802	.754	.759	.751	.740	.734	.750

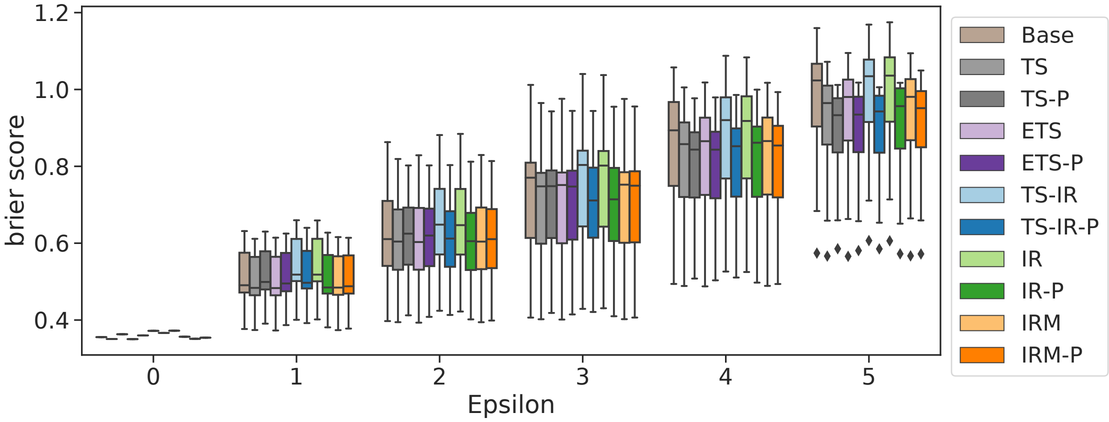


Figure 1: Brier score for Resnet50 trained on Imagenet

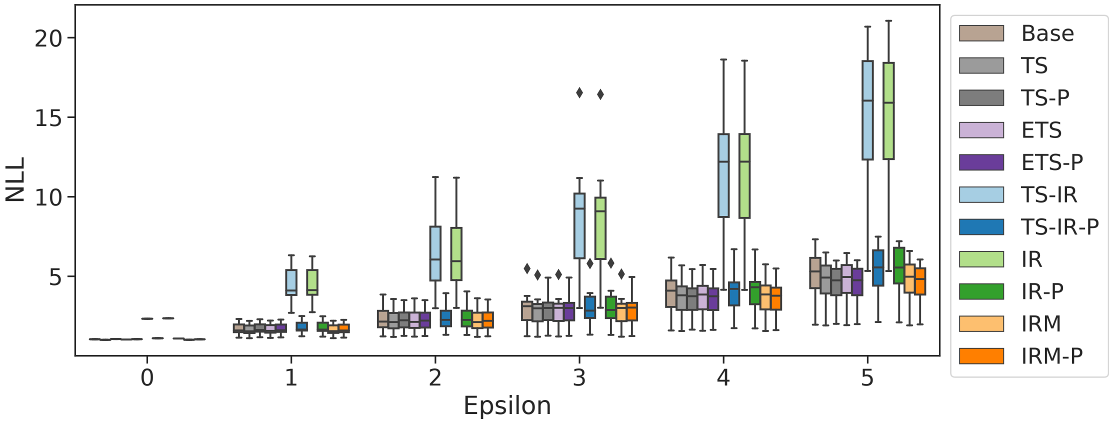


Figure 2: NLL for Resnet50 trained on Imagenet

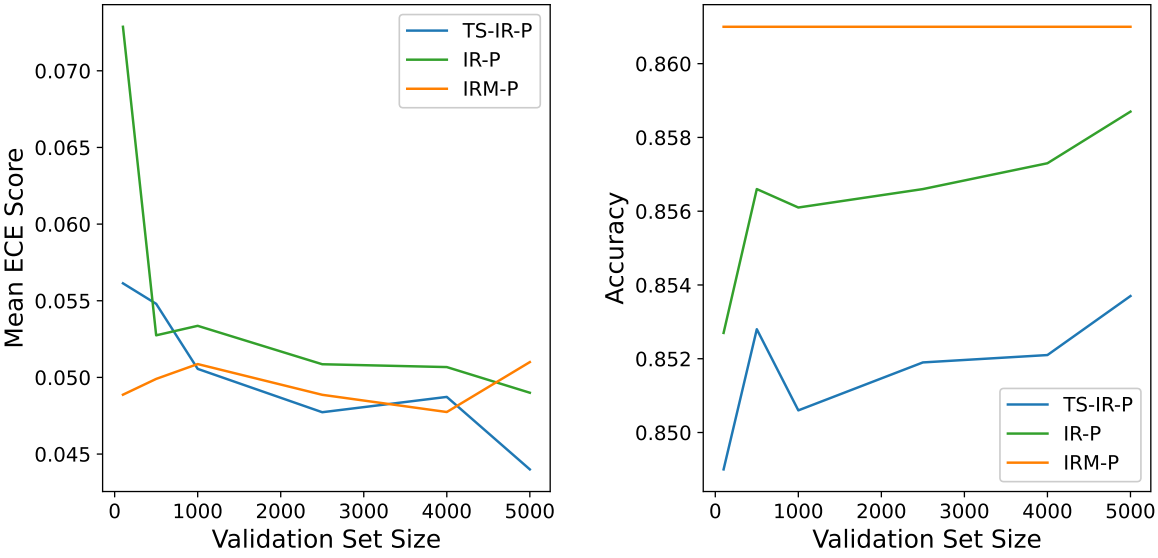
To further illustrate the benefit of our modelling approach for different post-calibration methods, we computed for each algorithm the difference in mean ECE between our approach (using a perturbed validation set) and the standard approach (using the unperturbed validation set). Table 7 highlights that our approach is beneficial for all post-calibration algorithms.

Table 7: ΔECE reveals that using a perturbed validation set for training improves performance across all methods for CIFAR-10 (higher is better).

	Δ TS	Δ ETS	Δ TS-IR	Δ IR	Δ IRM
CIFAR VGG19	0.661	0.622	0.706	0.718	**0.736**
CIFAR ResNet50	0.528	0.473	0.518	0.509	**0.575**
CIFAR DenseNet121	0.103	0.158	0.376	**0.472**	0.208
CIFAR MobileNetv2	0.281	0.113	0.428	**0.519**	0.266
ImgNet ResNet50	-0.022	0.365	**0.753**	0.740	0.428
ImgNet ResNet152	0.147	0.301	**0.778**	0.708	0.276
ImgNet VGG19	-1.044	-0.567	0.467	**0.762**	0.085
ImgNet Den.Net169	0.453	0.421	**0.795**	0.662	0.118
ImgNet Eff.NetB7	0.451	0.440	**0.705**	0.622	0.218
ImgNet Xception	0.110	0.253	**0.715**	0.221	-0.313
ImgNet MobileNetv2	0.304	0.348	0.644	**0.745**	0.356

3.3. Additional experiments

Size of validation set While both IRM-P and IR-P performed consistently well across baselines, a key difference is that IR-P is not accuracy preserving. In contrast, a model's accuracy remains unchanged after post-calibration with IRM-P. In the main paper, we show that the effect on the accuracy for IR-P is only marginal. To further investigate the robustness of IR-P in terms of accuracy, we assessed the effect of the size of the validation set on performance. Our results show, that in fact for small validation sets accuracy can substantially decrease for IR-P (Fig. 3 (b)). However, with increasing size of the validation set accuracy increases and ECE decreases (Fig. 3 (a)). This suggests that for sufficiently large validation set, IR-based methods benefit from their high expressiveness.



(a) Mean ECE w.r.t. size of validation set

(b) Accuracy w.r.t. size of validation set

Figure 3: Effect of the chosen size of the validation set on the mean expected calibration error and accuracy scores (CIFAR-10).

Base	TS	ETS	TS-IR	IR	IRM	TS-H	ETS-H	TS-IR-H	IR-H	IRM-H
0.323	0.158	0.152	0.173	0.176	0.167	0.102	**0.096**	0.112	0.127	0.114

Table 8: Mean expected calibration error across all test domain drift scenarios (affine transformations for CIFAR-10). Tuning was performed on the validation set and the perturbed validation set generated by applying the validation perturbations proposed in [1]. The latter is denoted by the suffix -H.

Type of validation perturbation Finally, we investigated the effect of the perturbation strategy used to generate a perturbed validation set. To this end, we assessed whether perturbing the validation set using image perturbations rather than the generic perturbations proposed in our work, could lead to similar results. To test this hypothesis, we used the validation perturbations *speckle noise*, *gaussian blur*, *spatter* and *saturate* introduced in [1] to generate a perturbed validation set. We then tuned all baselines on this validation set using a VGG19 model trained on CIFAR-10. Table 8 shows that this resulted in consistently worse calibration errors compared to the generic perturbation strategy proposed in the main paper. This suggests, that our algorithm can indeed yield a validation set that is representative of generic domain drift scenarios.

4. Guidelines

Based on our extensive experiments, we propose the following guidelines for practitioners:

- If a sufficiently large validation set is available and calibration for in-domain settings is of particular concern, we recommend using IR-P or TS-IR-P. This may result in changes in model accuracy.
- If the practitioner requires that the accuracy of the trained model remains unchanged or truly OOD scenarios are of particular concern, we recommend using IRM-P or ETS-P.

References

[1] Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. arXiv preprint arXiv:1903.12261 (2019)

[2] Kumar, A., Liang, P.S., Ma, T.: Verified uncertainty calibration. In: Advances in Neural Information Processing Systems. pp. 3792–3803 (2019)

[3] 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)

[4] Zadrozny, B., Elkan, C.: Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In: Icml. vol. 1, pp. 609–616. Citeseer (2001)